Six-factor model based on Chinese A-shares

林新凯 2016312010173

吕自立 2016302010145

杨宸宇 2016301550186

Investment Case Study 2018.12

投资学六因子案例

林新凯 2016312010173
吕自立 2016302010145
杨宸宇 2016301550186

摘要

本案例研究基于 Fama & French 1993 年在 The Journal of Finance 上发表的论文“Common Risk Factors in the Returns on Stocks and Bonds”，数据来源于万德数据库，调仓频率为月度，检验时间为 2018 年 1 月 1 日至 2017 年 12 月 31 日。其中我们主要考虑的因子包含了原始的 Fama 三因子，即市场风险因子，市值风险因子 SMB，账面市值比风险因子 HML 在内，度量波动的动量因子 MOM，盈利水平风险因子 RMW 与投资水平风险因子 CMA，整体发现因子较为有效，除了动量因子表现稍差以外其余因子都十分显著。

概述

本案例研究基于 Fama & French 1993 年在 The Journal of Finance 上发表的论文“Common Risk Factors in the Returns on Stocks and Bonds”，参考 Fama 的五因子模型以及 Carhart 的四因子模型，数据来源于万德数据库，调仓频率为月度，检验时间为 2018 年 1 月 1 日至 2017 年 12 月 31 日。其中我们主要考虑的因子包含了原始的 Fama 三因子，即市场风险因子，市值风险因子 SMB，账面市值比风险因子 HML 在内，度量波动的动量因子 MOM，盈利水平风险因子 RMW 与投资水平风险因子 CMA。经过研究，我们发现 HML 和 SMB、ROE 和 SMB 等因子间的相关性在研究期间比较高，从整体上来看，调整 R^2 较为理想，平均可以达到 94.53%，除去动量因子以外其他因子的 t 检验均显著，特别是 Market 和 SMB 因子，但在不同的 25 个组合中 β 和 SMB 相差较大。

研究方法

取样本期（2008-2017）的数据按照如下公式进行回归：

$$E_(r_i) - r_f = a_i + b_i[E(r_m) - r_f] + s_iE[SMB] + h_iE[HML] + d_iE[RMW] + c_iE[CMA] + m_iE[MOM]$$

其中，解释变量：

$[E(r_m) - r_f]$：市场指数（万德全 A）相对无风险利率（SHIBOR 隔夜利率）的超额收益

$[SMB]$：每个月按照总市值排列股票池（全部 A 股），依据上下四分位划分大小，即总市值最大的 25% 的股票作为大市值组合，总市值最小的 25% 的股票作为小市值组合，用小市值组合该月的个股收益率平均值减去大市值组合该月的个股收益率平均值，再减去无风险利率得到当月的 SMB 值，并按照这个方法计算样本期内每个月的所有 SMB 值.

$[HML]$：计算方式与 [SMB] 相同，只是将总市值换为市净率，用当月市净率最低（即账面市值比 BM 值最高）的投资组合的个股月度平均收益率减去当月市净率最高的投资组合的个股月度平均收益率，得到当月的 HML 值，再减去无风险利率得到 HML 的超额收益，即当月的 HML 值，并同样按照这个方法计算样本期内每个月的所有 HML 值.

$[RMW]$：计算方式与 [SMB] 相同，只是将总市值换为净资产收益率，用当月净资产收益率最低的投资组合的个股月度平均收益率减去当月净资产收益率最高的投资组合的个股月度平均收益率，得到当月的 RMW 值，再减去无风险利率得到 RMW 的超额收益，即当月的 RMW 值，并同样按照这个方法计算样本期内每个月的所有 RMW 值.

$[CMA]$：计算方式与 [SMB] 相同，只是将总市值换为资产增长率，用当月资产增长率最低的投资组合的个股月度平均收益率减去当月资产增长率最高的投资组合的个股月度平均收益率，得到当月的 CMA 值，再减去无风险利率得到 CMA 的超额收益，即当月的 CMA 值，并同样按照这个方法计算样本期内每个月的所有 CMA 值.

$[MOM]$：首先对前 11 期月度收益率进行排序得到前 30% 和后 30% 股票，计算两类的月度收益率差值，即得当月的 MOM 值，并同样按照这个方法计算样本期内每个月的所有 MOM 值.
被解释变量：

$E_(r_i) - r_f$：依据市值、账面价值/市值两个指标，按照五分位划分得到的 5 个大、中、小投资组合，再排列组合成 25 个交集，得到 25 个投资组合，计算他们的超额收益作为被解释变量.

数据

样本期：2008-01-01 至 2017-12-31

股票池：全部 A 股

剔除所有 ST 股票当月数据

剔除所有非正常交易股票当月数据

投资组合调仓频率：月度（每月末）

市场指数：用万德全 A 指数代替

无风险利率：用 SHIBOR 隔夜拆借利率的月度平均值代替

总市值（ME）：用公司的股权公平市场价值代替

账面市值比（BM）：用市净率代替

账面价值（BE）：用最新报告期资产负债表股东权益代替

净资产收益率：用万德的月度 ROE 代替

资产增长率：用万德的总资产增长率（相较年初）代替

报告内容

In [0]:
```
# Change the file path
import os
try:
    os.chdir(os.path.join(os.getcwd(), '..\OneDrive\工作\学习\第五学期\投资学'))
    print(os.getcwd())
except:
    pass
```

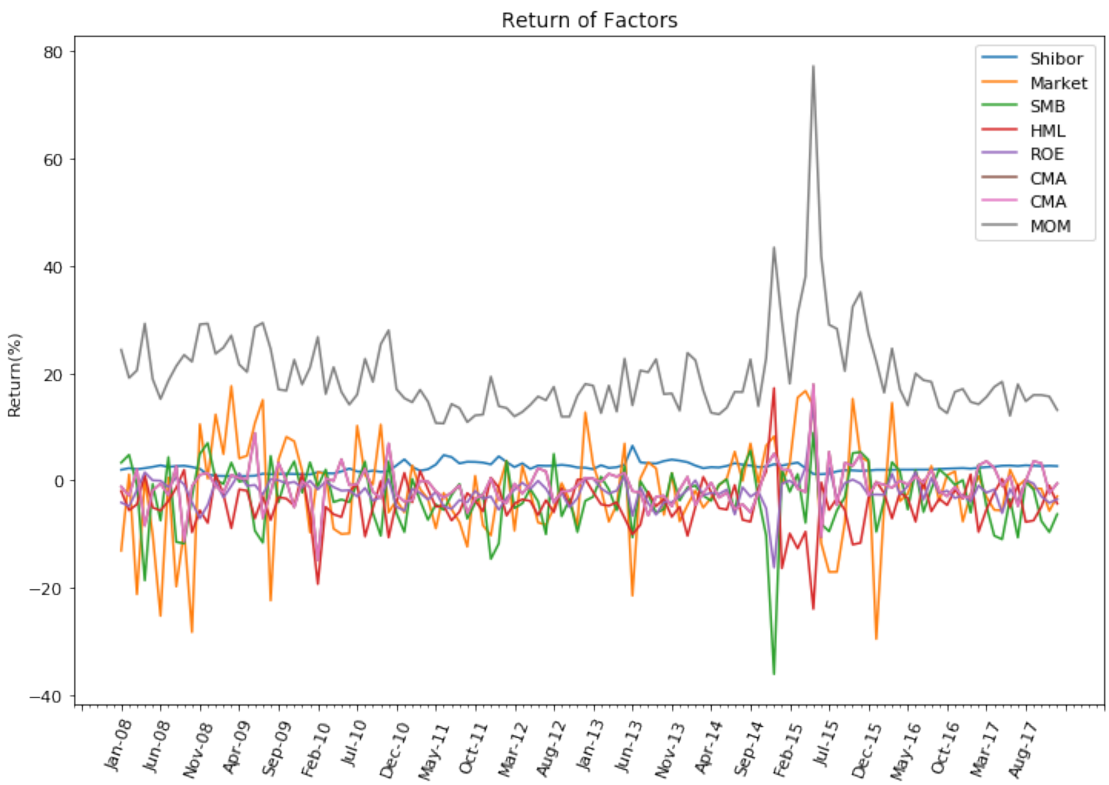
In [34]:
```
import os
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
plt.rcParams.update({"font.size": 11})
import numpy as np
import statsmodels.api as sm
```

```
         from IPython.display import display
         import seaborn as sns

In [35]: path = os.getcwd()

In [36]: data = pd.read_csv(
             open(
                 path + "\\FamaFrench.csv",
                 'r',
                 encoding = "utf-8"
             ),
             index_col = [0]
         )
         # data.index = pd.to_datetime(data.index, format = '%b-%y').strftime('%Y-%m')
```

1 收益率

可以看到，动量因子的收益率远高于其它因子，尤其是在 2015 年中。其余因子的收益率表现相差不大，均值都低于 0。

```
In [37]: plt.figure(figsize = (12, 8))
         ax = plt.axes()
         ax.xaxis.set_major_locator(ticker.MultipleLocator(5))
         ax.xaxis.set_minor_locator(ticker.MultipleLocator(1))
         plt.xticks(rotation = 70)
         plt.plot(data.index, data["Shibor"], label = "Shibor")
         plt.plot(data.index, data["Market"], label = "Market")
         plt.plot(data.index, data["SMB"], label = "SMB")
         plt.plot(data.index, data["HML"], label = "HML")
         plt.plot(data.index, data["ROE"], label = "ROE")
         plt.plot(data.index, data["CMA"], label = "CMA")
         plt.plot(data.index, data["CMA"], label = "CMA")
         plt.plot(data.index, data["MOM"], label = "MOM")
         plt.legend()
         plt.ylabel("Return(%)")
         plt.title("Return of Factors")

Out[37]: Text(0.5,1,'Return of Factors')
```



In [38]: data.describe()

```
Out[38]:
           Shibor      Market         SMB         HML         ROE         CMA  \
count  120.000000  120.000000  120.000000  120.000000  120.000000  120.000000
mean     2.391529   -1.855738   -3.077592   -4.464530   -2.133120   -0.963855
std      0.889648    8.989121    5.958622    4.553039    2.819122    3.803802
min      0.803582  -29.535931  -36.079260  -23.947787  -16.207890  -14.895867
25%      1.866313   -6.722316   -6.071562   -6.560288   -3.512013   -2.847319
50%      2.413980   -1.355685   -2.925407   -4.442765   -1.914440   -1.000553
75%      2.806922    2.844453    0.813304   -1.964928   -0.589128    0.741648
max      6.468176   17.584159    8.897063   17.213223   14.100332   17.898765

              MOM      ME0BP0      ME0BP1      ME0BP2     ...          ME3BP0  \
count  120.000000  120.000000  120.000000  120.000000     ...      120.000000
mean    19.761993    1.664841    1.117263    0.442514     ...        3.702431
std      8.325334   11.050989   11.104426   10.730139     ...       11.721207
min     10.611562  -26.104895  -31.835864  -33.176358     ...      -26.150575
```

```
25%      14.612304   -5.393821   -5.205359   -5.765252     ...      -3.062910
50%      17.552181    2.758905    2.282874    1.378960     ...       3.345745
75%      22.609008    8.841230    6.832669    6.706652     ...       9.730519
max      77.189281   31.084765   31.253123   25.181478     ...      63.250741

            ME3BP1      ME3BP2      ME3BP3      ME3BP4      ME4BP0      ME4BP1  \
count   120.000000  120.000000  120.000000  120.000000  120.000000  120.000000
mean      2.362642    1.780316    1.294371    0.762216    4.320273    1.809436
std      10.368887   10.244759   10.125552    9.594999   11.353038    9.085804
min     -26.386979  -29.065758  -29.133660  -26.452483  -21.352238  -23.547693
25%      -3.097955   -3.228491   -3.978126   -4.311305   -1.533493   -3.284972
50%       2.389918    2.167422    1.548886    0.767332    4.003287    2.385493
75%       7.411839    7.401030    6.484346    5.833401    8.323933    6.311672
max      38.905684   31.259587   24.768550   22.834147   62.828819   25.823137

            ME4BP2      ME4BP3      ME4BP4
count   120.000000  120.000000  120.000000
mean      1.265192    1.735443    1.100763
std       9.230414    8.721200    9.092975
min     -25.639417  -22.475910  -26.939990
25%      -2.977532   -2.524531   -3.831161
50%       2.117511    2.054107    1.382359
75%       5.552510    6.588335    5.345399
max      21.750587   22.480232   31.554630

[8 rows x 32 columns]
```

2 统计报告

2.1 因子间相关性

因子间的相关性比较大，这使得模型可能出现过拟合。其中 HML 和 SMB、ROE 和 SMB 等因子间的相关性特别高。如图为 SMB 和 HML 的图例。

```
In [39]: factors = list(data.columns[1:7])
         correlation = pd.DataFrame(index = factors, columns = factors)
         for factor_x in factors:
             for factor_y in factors:
```

```
        correlation.loc[factor_x, factor_y] = np.corrcoef(
            data[factor_x], data[factor_y]
        )[0][1]
correlation
```

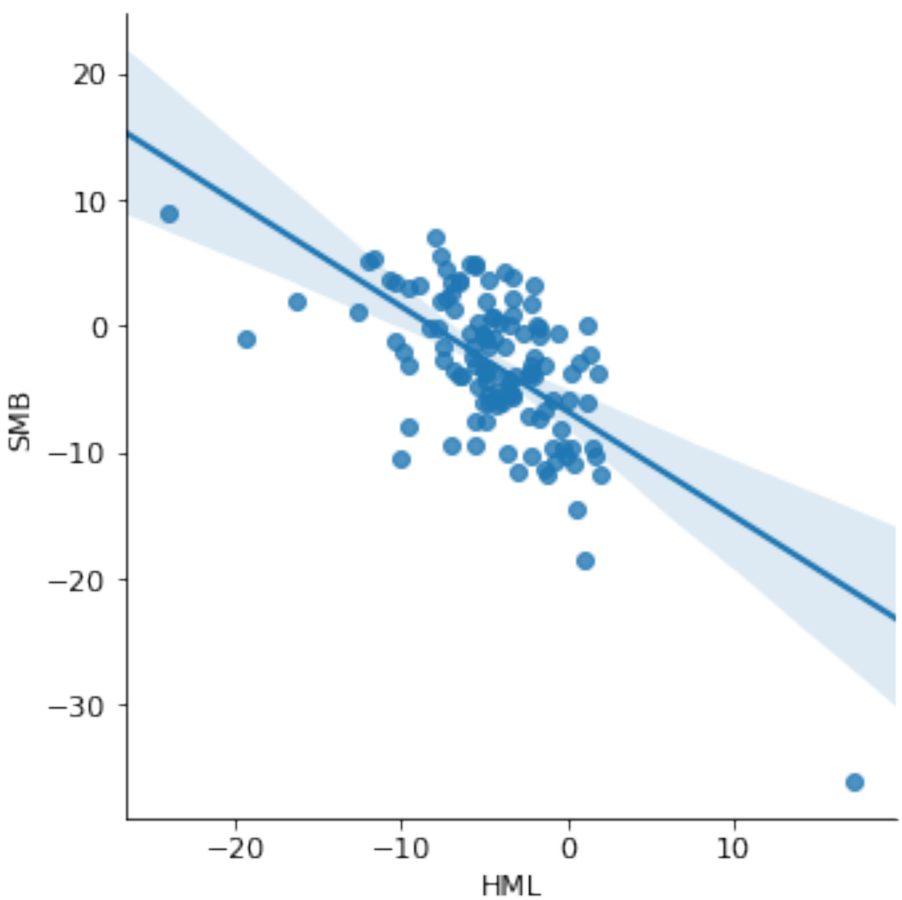
Out[39]:

	Market	SMB	HML	ROE	CMA	MOM
Market	1	0.209565	-0.0927207	0.14921	0.0843781	0.292963
SMB	0.209565	1	-0.634243	0.445343	0.101804	0.0489041
HML	-0.0927207	-0.634243	1	-0.554817	-0.230356	-0.298643
ROE	0.14921	0.445343	-0.554817	1	0.381226	0.312093
CMA	0.0843781	0.101804	-0.230356	0.381226	1	0.325905
MOM	0.292963	0.0489041	-0.298643	0.312093	0.325905	1

In [40]:

```
sns.lmplot("HML", "SMB", data)
```

Out[40]: <seaborn.axisgrid.FacetGrid at 0x1848974aa20>



2.2 因子参数

在 25 个分组的被解释变量中，因子的系数值相差不大，模型对不同风格的投资组合的解释性比较强。

```
In [41]: for factor in factors:
             parameters = pd.DataFrame(
                 index = ["ME" + str(i) for i in range(5)],
                 columns = ["BP" + str(i) for i in range(5)]
             )
             parameters.index.name = "Parameters of " + factor
             for i in range(5):
                 for j in range(5):
                     y = list(data["ME" + str(i) + "BP" + str(j)])
                     x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
                     x = sm.add_constant(x)
                     result = sm.OLS(y, x).fit()
                     parameters.iloc[i, j] = result.params[factor]
             display(parameters)
```

Parameters of Market	BP0	BP1	BP2	BP3	BP4
ME0	0.973124	0.983065	0.951422	0.994599	0.97529
ME1	1.00197	0.955012	0.989937	0.991315	0.960902
ME2	0.937634	0.985608	0.996106	0.965257	1.01143
ME3	0.961281	0.955536	0.994559	1.02404	0.977843
ME4	1.03506	0.989141	1.02331	0.934456	0.989226

Parameters of SMB	BP0	BP1	BP2	BP3	BP4
ME0	0.688744	0.714717	0.770626	0.700231	0.699419
ME1	0.551344	0.61614	0.658785	0.594067	0.616434
ME2	0.46613	0.435887	0.45953	0.529243	0.504404
ME3	0.330681	0.341306	0.361262	0.32556	0.288235
ME4	-0.668081	-0.252607	-0.223159	-0.296529	-0.299492

	BP0	BP1	BP2	BP3	BP4

Parameters of HML

ME0	-0.26278	-0.276935	-0.162308	-0.125571	-0.0648696
ME1	-0.343722	-0.275033	-0.201705	-0.0894505	-0.0357298
ME2	-0.359325	-0.330413	-0.201201	-0.111165	-0.0188457
ME3	-0.464203	-0.343909	-0.183689	-0.159066	0.000528283
ME4	-1.48505	-0.201939	-0.103954	-0.0120891	0.311209

Parameters of ROE

	BP0	BP1	BP2	BP3	BP4
ME0	-0.285129	-0.225641	-0.293962	-0.214524	-0.286509
ME1	-0.126359	-0.133787	-0.246443	-0.26342	-0.273881
ME2	0.27079	0.0149832	-0.0506243	-0.164069	-0.180784
ME3	0.293409	0.0271374	0.0218246	-0.0803569	-0.0717324
ME4	-0.166685	-0.361995	-0.271425	-0.0192589	-0.221915

Parameters of CMA

	BP0	BP1	BP2	BP3	BP4
ME0	0.00758401	-0.117193	-0.110249	-0.126299	-0.211194
ME1	0.158363	-0.138066	-0.0895266	-0.114957	-0.212069
ME2	0.213959	-0.123631	-0.145988	-0.226035	-0.222074
ME3	0.0104104	-0.0299374	-0.186399	-0.155059	-0.136125
ME4	-0.420353	0.0984175	0.0822943	-0.139896	-0.123009

Parameters of MOM

	BP0	BP1	BP2	BP3	BP4
ME0	0.0255467	0.0502839	0.0158153	-0.0395347	-0.0356836
ME1	0.120557	0.0670917	0.0353972	0.0162832	0.000880258
ME2	0.260102	0.104457	0.066073	0.0713211	0.0171894
ME3	0.281403	0.117339	0.078533	0.0117596	0.0509579
ME4	0.105291	0.0407669	-0.0211754	0.103513	0.0329027

2.3 因子 t 值

因子的 t 值指出其统计显著性。大部分的因子都至少在部分投资组合上展现出显著性。尤其是 Market 和 SMB 因子。

```
In [42]: for factor in factors:
             tvalues = pd.DataFrame(
                 index = ["ME" + str(i) for i in range(5)],
                 columns = ["BP" + str(i) for i in range(5)]
             )
             tvalues.index.name = "t values of " + factor
             for i in range(5):
                 for j in range(5):
                     y = list(data["ME" + str(i) + "BP" + str(j)])
                     x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
                     x = sm.add_constant(x)
                     result = sm.OLS(y, x).fit()
                     tvalues.iloc[i, j] = result.tvalues[factor]
             display(tvalues)
```

t values of Market	BP0	BP1	BP2	BP3	BP4
ME0	29.8646	44.6171	42.2722	44.1569	41.61
ME1	31.9636	41.7388	45.8218	42.5176	44.7146
ME2	22.572	34.1663	36.8254	40.2031	46.67
ME3	26.528	33.9254	40.1527	42.5493	39.7577
ME4	21.9876	38.0984	39.3933	39.147	58.2079

t values of SMB	BP0	BP1	BP2	BP3	BP4
ME0	10.939	16.7875	17.7198	16.0888	15.4431
ME1	9.10245	13.9362	15.7812	13.1864	14.8453
ME2	5.80734	7.81989	8.79203	11.4079	12.0451
ME3	4.72276	6.27126	7.54812	7.00067	6.065
ME4	-7.3447	-5.03531	-4.44596	-6.42893	-9.12021

t values of HML	BP0	BP1	BP2	BP3	BP4
ME0	-3.00392	-4.6817	-2.68613	-2.07656	-1.03089
ME1	-4.0843	-4.47736	-3.47767	-1.42905	-0.619308
ME2	-3.22204	-4.26636	-2.77063	-1.72461	-0.323905

```
ME3             -4.77165 -4.54808 -2.76232  -2.46183  0.00800064
ME4             -11.7506 -2.89717 -1.49062 -0.188642     6.82094
```

```
                      BP0       BP1       BP2       BP3       BP4
t values of ROE
ME0             -2.29255  -2.68303  -3.42184  -2.49525   -3.2025
ME1             -1.05608  -1.53191  -2.98861  -2.96002  -3.33903
ME2              1.70788  0.136077 -0.490331  -1.79032  -2.18549
ME3              2.12136  0.252426  0.230844 -0.874756 -0.764109
ME4            -0.927675  -3.65291   -2.7375 -0.211378  -3.42106
```

```
                      BP0       BP1      BP2      BP3      BP4
t values of CMA
ME0              0.09579  -2.18905 -2.01599 -2.30772 -3.70833
ME1              2.07917  -2.48342 -1.70549  -2.0292 -4.06144
ME2              2.11983  -1.76381 -2.22123 -3.87459 -4.21726
ME3             0.118237 -0.437446 -3.09714 -2.65158 -2.27784
ME4             -3.67501    1.5601  1.30383   -2.412 -2.97891
```

```
                     BP0      BP1       BP2       BP3        BP4
t values of MOM
ME0              0.66749  1.94298  0.598244  -1.49434   -1.29614
ME1              3.27428  2.49643   1.39494  0.594588  0.0348739
ME2              5.33091  3.08284   2.07963   2.52904   0.675277
ME3              6.61154  3.54683   2.69934  0.415997    1.76394
ME4              1.90424  1.33683 -0.694017   3.69194    1.64831
```

2.4 因子的 R^2

R^2 统计模型对数据的解释性，可以看到模型的解释性良好，在不同被解释变量中的平均值高达 94.8%。最好的有 97.29%，最差的也有 94.16%。

```
In [43]: rsquared = pd.DataFrame(
             index = ["ME" + str(i) for i in range(5)],
             columns = ["BP" + str(i) for i in range(5)]
```

```
    )
    rsquared.index.name = "R square of Regression"
    for i in range(5):
        for j in range(5):
            y = list(data["ME" + str(i) + "BP" + str(j)])
            x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
            x = sm.add_constant(x)
            result = sm.OLS(y, x).fit()
            rsquared.iloc[i, j] = result.rsquared
    display(rsquared)
```

	BP0	BP1	BP2	BP3	BP4
R square of Regression					
ME0	0.932553	0.969457	0.965867	0.965743	0.960922
ME1	0.941684	0.964263	0.968883	0.960963	0.964629
ME2	0.905969	0.944476	0.948719	0.956993	0.965179
ME3	0.925854	0.942757	0.954651	0.956172	0.949026
ME4	0.866619	0.936654	0.938558	0.941882	0.972901

```
In [44]: sum(list(rsquared.mean()))/5
```

```
Out[44]: 0.9480548778434097
```

```
In [45]: max(list(rsquared.max()))
```

```
Out[45]: 0.9729007171282669
```

```
In [46]: min(list(rsquared.max()))
```

```
Out[46]: 0.941683652490355
```

2.5

```
In [47]: rsquared_adj = pd.DataFrame(
             index = ["ME" + str(i) for i in range(5)],
             columns = ["BP" + str(i) for i in range(5)]
         )
         rsquared_adj.index.name = "Adjusted R square of Regression"
         for i in range(5):
```

```
            for j in range(5):
                y = list(data["ME" + str(i) + "BP" + str(j)])
                x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
                x = sm.add_constant(x)
                result = sm.OLS(y, x).fit()
                rsquared_adj.iloc[i, j] = result.rsquared_adj
        display(rsquared_adj)
```

```
                                     BP0       BP1       BP2       BP3  \
Adjusted R square of Regression
ME0                             0.928972  0.967835  0.964055  0.963924
ME1                             0.938587  0.962365  0.967231   0.95889
ME2                             0.900976  0.941528  0.945996   0.95471
ME3                             0.921917  0.939718  0.952243  0.953845
ME4                             0.859537  0.933291  0.935296  0.938796

                                     BP4
Adjusted R square of Regression
ME0                             0.958847
ME1                              0.96275
ME2                              0.96333
ME3                             0.946319
ME4                             0.971462
```

```
In [48]: sum(list(rsquared_adj.mean()))/5

Out[48]: 0.9452967297642989

In [49]: max(list(rsquared_adj.max()))

Out[49]: 0.9714618171527766

In [50]: min(list(rsquared_adj.max()))

Out[50]: 0.9385872092597544
```

2.6 回归报告

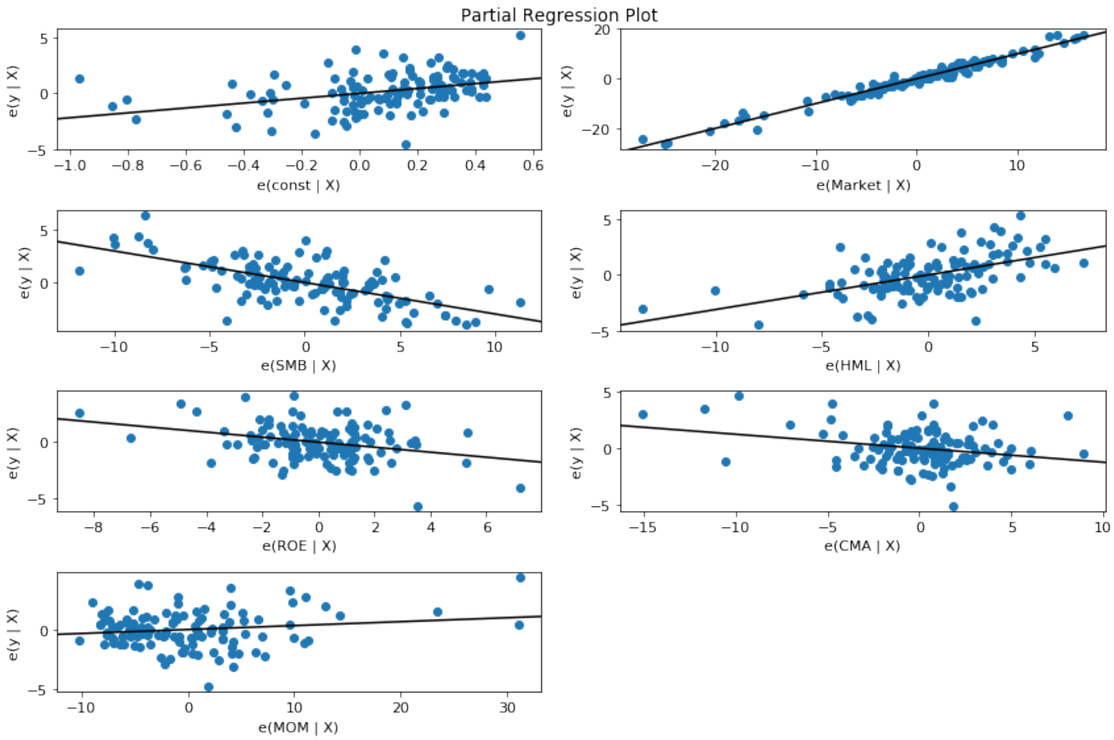
取其中一个被解释变量展示回归结果。可以看到，除了动量因子以外的所有银子，在统计上都是显著的 (t>2)。JB 检验和 DW 检验的效果也十分好。

```
In [51]: y = list(data["ME4BP4"])
         x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
         x = sm.add_constant(x)
         result = sm.OLS(y, x).fit()
         print(result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.973
Model:                            OLS   Adj. R-squared:                  0.971
Method:                 Least Squares   F-statistic:                     676.1
Date:                Wed, 19 Dec 2018   Prob (F-statistic):           4.62e-86
Time:                        21:02:59   Log-Likelihood:                -218.18
No. Observations:                 120   AIC:                             450.4
Df Residuals:                     113   BIC:                             469.9
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.1620      0.489      4.418      0.000       1.192       3.132
Market         0.9892      0.017     58.208      0.000       0.956       1.023
SMB           -0.2995      0.033     -9.120      0.000      -0.365      -0.234
HML            0.3112      0.046      6.821      0.000       0.221       0.402
ROE           -0.2219      0.065     -3.421      0.001      -0.350      -0.093
CMA           -0.1230      0.041     -2.979      0.004      -0.205      -0.041
MOM            0.0329      0.020      1.648      0.102      -0.007       0.072
==============================================================================
Omnibus:                        2.318   Durbin-Watson:                   2.234
Prob(Omnibus):                  0.314   Jarque-Bera (JB):                2.018
Skew:                           0.083   Prob(JB):                        0.365
Kurtosis:                       3.613   Cond. No.                         77.8
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

2.7 因子回归图

```
In [52]: fig = plt.figure(figsize = (12, 8))
         fig = sm.graphics.plot_partregress_grid(result, fig = fig)
```



总结

经过研究，在 2007 年 1 月 1 日到 2017 年 12 月 31 日所有 A 股除去 ST 和非政策交易股票的数据集中，我们发现 HML 和 SMB、ROE 和 SMB 等因子间的相关性在研究期间比较高，从整体上来看，通过 R^2 观察统计模型对数据的解释性，可以看到模型的解释性良好，在不同被解释变量中的 R^2 平均值高达 94.8%。最好的有 97.29%，最差的也有 94.16%。调整 R^2 也较为理想，平均可以达到 94.53%，除去动量因子以外其他因子的 t 检验均显著，特别是 Market 和 SMB 因子，JB 检验和 DW 检验的效果也十分好。但在不同的 25 个组合中 β 和 SMB 相差较大。

参考文献

[1]Common Risk Factors in The Returns On Stocks and Bonds, Eugene F. Fama a & Kenneth R. French, Journal of Financial Economics 33 (1993) 3-56. North-Holland.

[2]On Persistence in Mutual Fund Performances, Carhart, The Journey of Finance, NO.1, March 1997.

[3]A Five-Factor Asset Pricing Model, Eugene F. Fama a & Kenneth R. French, Journal of Financial Economics 116 (2015) 1–22.

[4]Anomalies in Chinese A Shares, Jason Hsu & Vivek Viswanathan & Michael Wang & Phillip Wool.

[5]Size, Value, and Momentum in International Stock Returns, Fama& Eugene F.& Kenneth R. French., Journal of Financial Economics 105, no. 3 (2012): 457-472.

[6]International Tests of A Five-Factor Asset Pricing Model, Fama& Eugene F.& Kenneth R. French.

[7]Profitability, Investment and Average Returns, Fama & Eugene F.& Kenneth R. French., Journal of Financial Economics 82, no. 3 (2006): 491-518.

数据处理部分代码 (Python)

```
In [1]: import os
        import pandas as pd
        PYk+knimport datetime as dt
        from scipy import stats
        import statsmodels.api as sm
        import matplotlib.pyplot as plt
        import seaborn as sns

        import WindPy as w
        from WindPy import *
        w.start()

        path = os.getcwd()
        start = "2008-01-01"
        end = "2017-12-31"

        def months_list(start = start, end = end):
            '''
            参数:
                start: 开始日期 ("YYYY-MM-DD")。(str)
                end: 结束日期 ("YYYY-MM-DD")。(str)
            返回:
                样本期的月份列表。(string list)
            '''
            file_path = path + r"/months.csv"
            if os.path.isfile(file_path):
                months = pd.read_csv(
                    open(file_path, 'r', encoding = 'utf-8'),
                    index_col = [0]
                )[12:]
                months_list = list(months["Month"])
                months_list = [x[:7] for x in months_list]
            else:
                months = w.tdays(start, end, "Period=M", usedf = True)[1]
                months.columns = ["Month"]
```

```python
        months.to_csv(file_path)
        months_list = list(months["Month"])
        months_list = [x.strftime('%Y-%m') for x in months_list]
    return months_list

months_list = months_list()

def market(
    start = start,
    end = end,
    hs300 = False,
    windA = True,
):
    '''
    参数:
        start: 开始日期 ("YYYY-MM-DD")。(str)
        end: 结束日期 ("YYYY-MM-DD")。(str)
        hs300: 是否用沪深 300 代表市场指数。(bool)
        windA: 用万德全 A 代表市场指数。(bool)
    返回:
        市场指数在样本期内的表现。(pd.DataFrame)
        index: Month, 日期 (YYYY-MM-DD)。(string)
        column: Market, 当月涨跌幅 (%)。(float)
    '''
    file_path = path + r"/market.csv"
    if os.path.isfile(file_path):
        market = pd.read_csv(
            open(file_path, 'r', encoding = 'utf-8'),
            index_col = [0]
        )[12:]
    else:
        if hs300:
            market_code = "000300.SH"
        elif windA:
            market_code = "881001.WI"
        market = w.wsd(
            market_code,
```

```
            "pct_chg",
            start,
            end,
            "Period=M",
            usedf = True
        )[1]
        market.index = pd.to_datetime(market.index).strftime("%Y-%m")
        market.index.name = "Month"
        market.columns = ["Market"]
        market.dropna(inplace = True) # 剔除缺失值
        market.to_csv(file_path)
    return market

market = market()

def shibor(start = start, end = end):
    '''
    参数:
        start: 开始日期 ("YYYY-MM-DD")。(str)
        end: 结束日期 ("YYYY-MM-DD")。(str)
    返回:
        无风险利率, 以 SHIBOR 隔夜利率的月平均值代表。(pd.DataFrame)
        index: Month, 日期 (YYYY-MM-DD)。(pd.datetime)
        column: Shibor, 当月涨跌幅 (%)。(float)
    '''
    file_path = path + r"/shibor.csv"
    if os.path.isfile(file_path):
        shibor = pd.read_csv(
            open(file_path, 'r', encoding = 'utf-8'),
            index_col = [0]
        )[12:]
    else:
        shibor = w.wsd(
            "SHIBORON.IR",
            "close",
            start,
            end,
```

```python
            "",
            usedf = True
        )[1]
        shibor.index = pd.to_datetime(shibor.index).strftime('%Y-%m')
        shibor = pd.DataFrame(shibor.groupby(shibor.index)["CLOSE"].mean())
        shibor.index.name = "Month"
        shibor.columns = ["Shibor"]
        shibor.dropna(inplace = True) # 剔除缺失值
        shibor.to_csv(file_path)
    return shibor

shibor = shibor()

def all_data(
    start = start,
    end = end,
    universe = "A",
    trading_only = True,
    non_ST_only = True
):
    '''
    参数:
        start: 开始日期 ("YYYY-MM-DD")。(str)
        end: 结束日期 ("YYYY-MM-DD")。(str)
        universe: 股票池，沪深 300('hs300') 或全部 A 股 ('A')。(str)
        trading_only: 是否只保留正常交易的股票。(bool)
        non_ST_only: 是否只保留非 ST 股票。(bool)
    返回:
        指定样本期的全部数据。(pd.DataFrame)
        index: Month, 日期 (YYYY-MM)。(str)
        columns:
            Code, 股票代码。(str)
            Name, 股票简称。(str)
            Return, 当月涨跌幅 (%)。(float)
            ME, 当月总市值。(float)
            Book, 当月账面价值。(float)
            Price, 当月股价。(float)
```

```python
        BP, 当月账面市值比。(float)
        Asset, 当月账面价值。(float)
        ROE, 权益回报。(float)
        ST, 是否为 ST 股票。(str)
'''

file_path = path + r"/data.csv"

if os.path.isfile(file_path):
    data = pd.read_csv(
        open(
            file_path,
            'r',
            encoding = 'utf-8'
        ),
        index_col = [0]
    )
    if trading_only:
        data = data.dropna() # 剔除缺失值
    if non_ST_only:
        data = data[data['ST'] == '否'] # 剔除 ST 股票
    data["Asset"] = data["Asset"].astype("str")
    data["Asset"] = [''.join(x.split(",")) for x in list(data["Asset"])]
    data["Asset"] = data["Asset"].astype("float")

else:

    if universe == "hs300":
        stocks_list = list(w.wset(
            "sectorconstituent",
            "date="+end+";windcode=000300.SH",
            usedf = True
        )[1].sample(100)['wind_code']) # ".sample(100)" 仅测试用

    elif universe == "A":
        stocks_list = list(
            w.wset(
```

```python
                    "sectorconstituent",
                    "date="+end+";sectorid=a001010100000000",
                    usedf = True
                )[1]['wind_code'] # ".sample(100)" 仅测试用
            )

        data = pd.DataFrame()

        for stock in stocks_list:
            stock_data = w.wsd(
                stock,
                '''trade_code,pct_chg,ev,roe,yoyassets,trade_status
                ,riskwarning''',
                start,
                end,
                '''unit=1;ruleType=3;period=2;returnType=1;index=000001.SH;
                Period=M;Fill=Previous''',
                usedf = True
            )[1]
            stock_data.index = pd.to_datetime(stock_data.index)\
                                    .strftime("%Y-%m")
            data = data.append(stock_data)

        data.index.name = "Month"
        data.columns = [
            "Code", "Return",
            "ME", "ROE",
            "Asset", "Status", "ST"
        ]

        data.to_csv(file_path)

    return data

data = all_data()

def excess(data):
```

```python
    '''
    参数:
        data: 要操作的数据表。(pd.DataFrame)
    返回:
        添加了无风险利率和超额收益列的原数据表。(pd.DataFrame)
    '''
    col_list = list(data.columns)
    data["Shibor"] = list(shibor["Shibor"])
    for column in col_list:
        data[column] = data[column] - data["Shibor"]
    return data[col_list]

def value_weighted_data(data):
    '''
    参数:
        data: 要操作的数据表。(pd.DataFrame)
    返回:
        将 Return 列替换为按 ME 列（市值）加权后的 Return。(pd.DataFrame)
    '''
    total_ME = data.sum(axis = 0)["ME"]
    data["Weight"] = data["ME"] / total_ME
    data["Return"] = data["Return"] * data["Weight"]
    return data

def monthly_return(factor, value_weighted = True):
    '''
    参数:
        factor: 因子指标名。(str)
        value_weighted: 是否将收益市值加权。(bool)
    返回:
        每个月按照因子排列的大小投资组合的收益。(pd.DataFrame)
    '''
    small_ret_list, big_ret_list = [], []
    for month, monthly_data in data.groupby(data.index):
        sort = monthly_data.sort_values(by = factor)
        small = sort[:round(len(monthly_data)/3)]
        big = sort[-round(len(monthly_data)/3):]
```

```python
        if value_weighted:
            small_ret = value_weighted_data(small).sum(axis = 0)["Return"]
            big_ret = value_weighted_data(big).sum(axis = 0)["Return"]
        else:
            small_ret = small.sum(axis = 0)["Return"]/len(small)
            big_ret = big.sum(axis = 0)["Return"]/len(big)
        small_ret_list.append(small_ret)
        big_ret_list.append(big_ret)
    monthly_return = pd.DataFrame(index = months_list)
    monthly_return["Small " + factor] = small_ret_list
    monthly_return["Big " + factor] = big_ret_list
    return monthly_return

def MKT(excess_return = True):
    '''
    参数:
        excess_return: 是否计算超额收益。(bool)
    返回:
        指定 Fama French 三因素模型的市场因子。(pd.DataFrame)
    '''
    MKT = market
    MKT.columns = ["MKT"]
    if excess_return:
        MKT = excess(MKT)
    return MKT

MKT = MKT()

def SMB(excess_return = True):
    '''
    参数:
        excess_return: 是否计算超额收益。(bool)
    返回:
        指定 Fama French 三因素模型中的 HML 因子。(pd.DataFrame)
    '''
    data = monthly_return('ME')
    data["SMB"] = data["Small ME"] - data["Big ME"]
```

```python
    if excess_return:
        data = excess(data)
    return data[["SMB"]]

SMB = pd.read_csv(path + r"/SMB.csv", index_col = [0])

SMB.to_csv(path + r"SMB.csv")

def HML(excess_return = True):
    '''
    参数:
        excess_return: 是否计算超额收益。(bool)
    返回:
        指定 Fama French 三因素模型中的 HML 因子。(pd.DataFrame)
    '''
    data = monthly_return('BP')
    data["HML"] = data["Big BP"] - data["Small BP"]
    if excess_return:
        data = excess(data)
    return data[["HML"]]

HML = HML()

def ROE(excess_return = True):
    '''
    参数:
        excess_return: 是否计算超额收益。(bool)
    返回:
        指定 Fama French 模型中的 ROE 因子。(pd.DataFrame)
    '''
    data = monthly_return("ROE")
    data["ROE"] = data["Big ROE"] - data["Small ROE"]
    if excess_return:
        data = excess(data)
    return data[["ROE"]]

ROE = ROE()
```

```python
def CMA(excess_return = True):
    '''
    参数:
        excess_return: 是否计算超额收益。(bool)
    返回:
        指定 Fama French 模型中的 CMA 因子。(pd.DataFrame)
    '''
    data = monthly_return("Asset")
    data["CMA"] = data["Big Asset"] - data["Small Asset"]
    if excess_return:
        data = excess(data)
    return data[["CMA"]]

CMA = CMA()

def MOM(excess_return = True):
    '''
    参数:
        excess_return: 是否计算超额收益。(bool)
    返回:
        指定 Fama French 模型中的 MOM 因子。(pd.DataFrame)
    '''
    data = monthly_return("Return")
    data["MOM"] = data["Big Return"] - data["Small Return"]
    if excess_return:
        data = excess(data)
    return data [["MOM"]]

MOM = MOM()

def Y(excess_return = True):
    '''
    参数:
        excess_return: 是否计算超额收益。(bool)
    返回:
        被解释变量。(pd.DataFrame)
```

```
    '''
    Y = {}
    for i in range(5):
        for j in range(5):
            Y["ME" + str(i) + "BP" + str(j)] = []
    for month, monthly_data in data.groupby(data.index):
        sort_ME = monthly_data.sort_values(by = "ME")
        sort_BP_ME = monthly_data.sort_values(by = "BP")
        length = round(len(monthly_data)/5)
        for i in range(5):
            for j in range(5):
                ME_list = list(sort_ME[i*length:(i+1)*length]["Code"])
                BP_ME_list = list(sort_BP_ME[j*length:(j+1)*length]["Code"])
                stock_list = [x for x in ME_list if x in BP_ME_list]
                portfolio = monthly_data[monthly_data["Code"]\
                                                     .isin(stock_list)]
                portfolio_ret = value_weighted_data(portfolio)\
                                              .sum(axis = 0)["Return"]
                Y["ME" + str(i) + "BP" + str(j)].append(portfolio_ret)
    return pd.DataFrame(Y, index = months_list)

Y = Y()

FamaFrench = pd.DataFrame()
FamaFrench["Shibor"] = shibor["Shibor"]
FamaFrench["Market"] = MKT["MKT"]
#FamaFrench["SMB"] = SMB["SMB"]
FamaFrench["HML"] = HML["HML"]
FamaFrench["ROE"] = ROE["ROE"]
FamaFrench["CMA"] = CMA["CMA"]
FamaFrench["MOM"] = MOM["MOM"]
FamaFrench = pd.concat(
    [FamaFrench, Y],
    axis = 1,
    sort = False
)
```

```
FamaFrench.to_csv(path + r"/FamaFrench.csv")


FamaFrench = pd.read_csv(
    open(
        path + r"/FamaFrench.csv",
        'r',
        encoding = 'utf-8'
    ),
    index_col = [0]
)
```

数据分析部分代码 (Python)

```python
import os
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
plt.rcParams.update({"font.size": 11})
import numpy as np
import statsmodels.api as sm
from IPython.display import display
import seaborn as sns

path = os.getcwd()

data = pd.read_csv(
    open(
        path + "\\FamaFrench.csv",
        'r',
        encoding = "utf-8"
    ),
    index_col = [0]
)
data.index = pd.to_datetime(data.index, format = '%b-%y').strftime('%Y-%m')


# 收益率
# 可以看到，动量因子的收益率远高于其它因子，尤其是在 2015 年中。
plt.figure(figsize = (12, 8))
ax = plt.axes()
ax.xaxis.set_major_locator(ticker.MultipleLocator(5))
ax.xaxis.set_minor_locator(ticker.MultipleLocator(1))
plt.xticks(rotation = 70)
plt.plot(data.index, data["Shibor"], label = "Shibor")
plt.plot(data.index, data["Market"], label = "Market")
plt.plot(data.index, data["SMB"], label = "SMB")
plt.plot(data.index, data["HML"], label = "HML")
plt.plot(data.index, data["ROE"], label = "ROE")
```

```python
plt.plot(data.index, data["CMA"], label = "CMA")
plt.plot(data.index, data["CMA"], label = "CMA")
plt.plot(data.index, data["MOM"], label = "MOM")
plt.legend()
plt.ylabel("Return(%)")
plt.title("Return of Factors")


data.describe()


# 统计报告

# 因子间相关性



factors = list(data.columns[1:7])
correlation = pd.DataFrame(index = factors, columns = factors)
for factor_x in factors:
    for factor_y in factors:
        correlation.loc[factor_x, factor_y] = np.corrcoef(
            data[factor_x], data[factor_y]
        )[0][1]
correlation

sns.lmplot("HML", "SMB", data)


# 因子参数
# 其中系数相对较大的有 $\beta$ 和 SMB。

for factor in factors:
    parameters = pd.DataFrame(
        index = ["ME" + str(i) for i in range(5)],
        columns = ["BP" + str(i) for i in range(5)]
    )
    parameters.index.name = "Parameters of " + factor
```

```
    for i in range(5):
        for j in range(5):
            y = list(data["ME" + str(i) + "BP" + str(j)])
            x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
            x = sm.add_constant(x)
            result = sm.OLS(y, x).fit()
            parameters.iloc[i, j] = result.params[factor]
    display(parameters)


for factor in factors:
    tvalues = pd.DataFrame(
        index = ["ME" + str(i) for i in range(5)],
        columns = ["BP" + str(i) for i in range(5)]
    )
    tvalues.index.name = "t values of " + factor
    for i in range(5):
        for j in range(5):
            y = list(data["ME" + str(i) + "BP" + str(j)])
            x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
            x = sm.add_constant(x)
            result = sm.OLS(y, x).fit()
            tvalues.iloc[i, j] = result.tvalues[factor]
    display(tvalues)



rsquared = pd.DataFrame(
    index = ["ME" + str(i) for i in range(5)],
    columns = ["BP" + str(i) for i in range(5)]
)
rsquared.index.name = "R square of Regression"
for i in range(5):
    for j in range(5):
        y = list(data["ME" + str(i) + "BP" + str(j)])
        x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
        x = sm.add_constant(x)
```

```python
        result = sm.OLS(y, x).fit()
        rsquared.iloc[i, j] = result.rsquared
display(rsquared)

rsquared_adj = pd.DataFrame(
    index = ["ME" + str(i) for i in range(5)],
    columns = ["BP" + str(i) for i in range(5)]
)
rsquared_adj.index.name = "Adjusted R square of Regression"
for i in range(5):
    for j in range(5):
        y = list(data["ME" + str(i) + "BP" + str(j)])
        x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
        x = sm.add_constant(x)
        result = sm.OLS(y, x).fit()
        rsquared_adj.iloc[i, j] = result.rsquared_adj
display(rsquared_adj)

# 回归报告
# 取其中一个被解释变量展示回归结果。


y = list(data["ME4BP0"])
x = data.loc[:, ["Market", "SMB", "HML", "ROE", "CMA", "MOM"]]
x = sm.add_constant(x)
result = sm.OLS(y, x).fit()
print(result.summary())

# 因子回归图
fig = plt.figure(figsize = (12, 8))
fig = sm.graphics.plot_partregress_grid(result, fig = fig)
```

Month	Shibor	Market	HML	ROE	CMA	MOM	ME0BP0	ME0BP1	ME0BP2	ME0BP3
2008-01	1.98321818	-13.071154	-1.9839089	-4.1115219	-1.1606112	24.3305687	-9.4969724	-10.320073	-8.9967483	-8.3964115
2008-02	2.2916125	1.13271686	-5.5769219	-4.8145673	-2.8728804	19.0991068	11.4576086	9.19327256	9.6459851	8.29347267
2008-03	2.08465714	-21.200326	-4.4393246	-1.8186451	2.08871639	20.4957582	-18.51633	-15.959303	-15.476359	-17.100119
2008-04	2.2890619	0.37896327	0.99046012	1.50859268	-8.3992346	29.2263357	-11.5521	-9.8680484	-12.550924	-8.9548492
2008-05	2.520375	-10.067207	-5.0481654	0.01686379	-1.8877719	18.8832018	-3.7705372	-1.5785582	-3.6949054	-1.0092947
2008-06	2.81831	-25.233117	-5.6301383	-0.0543491	-0.5810571	15.1709456	-25.620735	-24.440067	-22.93946	-28.864424
2008-07	2.50766957	0.53785363	-3.8207153	-1.2224459	-2.7779999	18.5943646	9.85035315	11.6582302	9.59556539	12.1656322
2008-08	2.66094286	-19.762012	-1.369344	0.58489846	2.67746749	21.2998415	-24.027222	-24.463867	-24.478031	-24.054532
2008-09	2.7379	-9.3544166	1.9682406	-0.7535367	-11.362224	23.3822356	-8.965124	-11.371063	-11.235016	-12.615507
2008-10	2.50318	-28.251809	-9.6468757	-4.074657	-1.0783783	22.1144772	-26.104895	-25.781057	-25.941688	-26.73201
2008-11	2.15235	10.4847038	-5.5702384	-7.0605481	1.0446781	29.0760514	23.5009793	21.4458925	22.4557028	18.012874
2008-12	1.12292174	0.33995056	-7.9175126	-4.5769442	1.40483967	29.218128	4.58257218	10.9237537	11.1854255	5.7071674
2009-01	0.85592	12.2882036	1.15637145	-0.1884888	-1.0452698	23.5959995	15.2319096	15.3655663	13.2131593	13.1402949
2009-02	0.82156	4.91162281	-2.62005	-3.1521027	-2.270187	24.7418109	10.4622895	5.53691672	5.3862676	7.44088654
2009-03	0.80358182	17.5841591	-8.8936249	-1.0164476	1.01946754	27.0012213	23.3688087	21.3475406	21.5907332	21.1302359
2009-04	0.80666667	4.11132664	-1.6771908	1.33877431	0.41776094	21.6028842	4.79554178	6.78636064	7.41891949	5.30949289
2009-05	0.80983333	4.60736812	-1.9167231	-1.023747	0.21694335	20.1968886	9.4930504	6.35377068	8.48503846	4.38523701
2009-06	0.87364545	10.7003999	-7.0734588	-0.8377384	8.82155674	28.5353409	10.3585107	4.03564587	3.18118064	4.378972
2009-07	1.25266522	14.9990027	-2.9794107	-2.6654306	-7.0169504	29.3506976	10.0292286	11.578991	10.2742238	8.63486018
2009-08	1.15700476	-22.370038	-7.3768774	0.25952381	-2.1685515	24.4972628	-15.893394	-13.002292	-14.943974	-18.184568
2009-09	1.20472273	3.86908444	-3.1069672	0.03351511	3.2688606	16.9550728	1.64811928	3.03571765	0.98011748	4.58402531
2009-10	1.238175	8.12341292	-3.3821225	-0.5473287	-0.0696566	16.7297368	11.2504136	11.8753076	11.9305391	11.7069744
2009-11	1.19369048	7.28944702	-4.708257	-0.2599579	-4.9990683	22.4874666	13.7653861	13.9238052	15.4839408	13.6705388
2009-12	1.17967826	1.78523985	1.26357349	-1.3346142	-0.1293745	17.8853681	-1.3029043	4.69616528	3.47793004	3.86389417
2010-01	1.11511	-9.6869116	-6.5776072	-0.0577038	-1.293168	20.9994245	-6.0863037	-1.5480145	-3.4980169	-3.4923624
2010-02	1.53203333	1.56793615	-19.298429	-1.672778	-14.895867	26.7261952	9.40550874	8.73060124	6.63500643	5.67623752
2010-03	1.34753913	1.20747934	-4.9334446	-0.2005222	0.37795965	16.1114256	8.12526438	5.3855422	6.39039737	5.06978928
2010-04	1.30440952	-8.9970818	-6.3104492	-1.2424373	-0.0925171	21.1001109	-10.409585	-7.4976507	-10.984498	-10.783206
2010-05	1.68171	-9.9847961	-6.8103661	-1.9035272	3.94558362	16.4751931	-11.019039	-7.8060422	-10.066122	-11.574414

ME0BP4	ME1BP0	ME1BP1	ME1BP2	ME1BP3	ME1BP4	ME2BP0	ME2BP1	ME2BP2	ME2BP3	ME2BP4
-5.169178	-7.5072157	-4.8010544	-7.0586463	-6.8029974	-4.0959815	-5.6450942	-6.1932396	-4.9734832	-4.0763534	-6.3098638
11.5622411	12.1602992	9.6469314	11.9907632	10.5921872	8.72414955	9.51994187	11.1632761	11.3521532	10.6067915	6.90199875
-21.109639	-18.981875	-16.95311	-19.674648	-18.965671	-20.343198	-14.131082	-18.562353	-19.609211	-19.430702	-21.63379
-11.963587	-7.6539438	-3.7368671	-10.458081	-9.3820354	-10.380507	-2.8663253	-2.7961606	-3.1291966	-5.5298494	-4.4697584
-6.0692952	-1.9845016	0.72045701	-4.7588053	-4.6716845	-4.0253358	1.74389169	-0.6700764	-2.9252119	-3.6778148	-6.9595635
-28.501908	-26.088239	-23.415928	-27.670344	-26.846371	-27.512535	-20.003638	-25.841554	-24.004815	-26.891446	-26.962953
11.0582956	13.0881382	9.22625031	10.8148101	11.5257822	10.9262504	10.8801004	11.4059823	10.4874972	8.85922116	9.18827606
-23.289401	-22.282486	-22.488529	-24.458997	-22.547396	-23.869639	-22.394877	-22.90159	-22.053613	-25.146554	-24.017793
-8.4961294	-9.3753159	-10.250247	-11.972359	-8.2209849	-7.4651713	-9.0900721	-7.3835776	-6.5550265	-9.486153	-8.6790864
-25.135855	-23.231572	-25.508535	-25.982898	-26.278172	-25.597099	-23.338518	-27.149125	-29.359963	-26.495668	-27.607934
17.0064903	23.1379725	23.2829353	21.5527177	19.7101141	19.1104767	24.125274	18.8882049	22.9634576	16.9361101	19.8709397
3.58204395	16.221079	11.2476351	8.69803806	8.53554785	3.36596222	14.0417187	9.48255509	3.12585092	3.56029845	3.749979
17.4232841	13.5767707	16.6244839	14.3873758	14.5092317	16.695559	17.3860448	14.6288467	18.5668131	19.2013931	17.7146973
6.02116718	9.35845211	6.89903983	10.5966739	7.09365481	9.23830006	10.4035909	7.52126065	13.1451225	9.51343749	11.1121767
21.9960164	27.2907849	23.9427478	23.0838253	23.216383	22.1068453	23.9478412	21.2556677	20.7446978	21.5866191	21.9190275
7.90401615	11.1874548	7.19479241	9.53817846	7.95774408	2.28158509	6.00667927	11.1516685	6.99365492	4.40724984	6.7313386
5.94527752	10.1216939	9.07967986	6.93784812	5.15946766	7.86460291	10.9419362	7.99315565	5.96213782	7.40254144	6.8127563
3.56376306	13.9325696	5.59789149	2.45039764	6.71790378	2.00037658	7.64810023	2.95378016	4.08804869	3.53588586	4.73085232
8.64231166	14.0243937	16.6176708	14.3174813	11.462307	10.6636999	12.6381816	15.4603659	13.7518294	13.3390558	16.8472574
-17.723478	-13.169785	-13.40854	-16.14482	-17.767034	-13.80901	-12.710574	-15.706991	-17.485502	-13.859127	-14.847193
3.59961748	1.81776808	3.62720678	5.48974604	4.15577578	2.77799089	19.6928865	2.76659728	3.72315616	0.82337276	1.33158804
12.457704	10.7280945	14.437277	11.7454104	12.8891009	11.2242902	9.46817377	12.0273789	11.2937571	10.7570274	10.7212223
11.5414209	16.8109064	13.7857309	17.4208275	17.98391	14.8157828	16.163377	14.8298796	13.1701377	15.7199745	16.8308761
4.27918096	1.15788719	3.50465891	3.45214812	3.91799796	1.53581087	1.02269904	3.55416553	3.45057389	3.37237232	1.09901508
-4.8260324	-5.8841222	-1.1007874	-1.5603941	-4.2600858	-2.4363351	-3.1130114	-3.8763961	1.49196431	-2.4530381	-5.1942819
8.38227396	8.07951378	8.66983838	7.92659304	5.87270988	5.06812599	4.79138018	8.63981256	7.02911784	7.11844146	5.27307427
3.88865624	9.74396768	6.60674515	5.66519633	1.6837983	2.66591152	8.53108541	4.13619977	3.85440172	2.25625206	3.32639105
-10.012687	-4.8005479	-5.8088577	-9.8320399	-11.026787	-10.407882	-6.0544784	-6.2675801	-7.8546107	-8.3975989	-11.305472
-11.494106	-8.0975671	-5.5320685	-8.0922595	-11.64958	-11.565298	-7.007913	-4.6498869	-9.1738466	-8.3879875	-10.126269

ME3BP0	ME3BP1	ME3BP2	ME3BP3	ME3BP4	ME4BP0	ME4BP1	ME4BP2	ME4BP3	ME4BP4
-4.1226997	-5.4306689	-5.0874796	-4.2911934	-4.8102377	-11.538205	-11.914051	-18.80572	-14.678281	-8.9637458
7.95045208	7.62278526	7.09546366	9.3777674	5.88875189	5.86867496	4.68733263	1.70052699	2.27969602	1.22623761
-19.407683	-20.929913	-17.699222	-20.728565	-18.2577	-18.915317	-19.334779	-19.977057	-19.341432	-19.643837
2.06202336	-0.1651336	-2.3771336	-2.2993797	0.01559389	8.68941859	2.91300329	4.4134109	11.3150455	7.67361204
-3.5769529	-3.888449	-2.0995606	-4.3672816	-7.425575	-3.7744046	-8.329447	-11.306421	-0.2387148	-6.5975511
-17.280196	-21.884139	-24.394743	-25.415234	-26.452483	-17.602372	-21.692132	-25.639417	-22.47591	-20.251941
9.46107282	7.34152381	8.03812488	7.08289447	8.98634665	4.23003795	3.4238496	3.64382497	1.52285311	1.49286402
-21.237796	-20.354761	-24.44979	-25.147682	-21.448197	-15.860479	-16.464913	-11.903999	-11.8965	-18.977051
-4.4846147	-7.9706703	-6.7648733	-6.0609038	-5.7926564	-2.0506947	-1.5626205	-6.9458199	2.40317632	1.37837347
-22.718203	-24.424067	-27.033576	-28.673987	-26.021285	-19.104617	-22.823841	-24.963553	-21.444341	-26.93999
24.9209751	19.282479	19.7596204	17.3894487	16.3915481	15.3880703	15.1438478	15.2205599	11.8254359	10.6629727
12.7067564	10.1510874	5.72146078	6.19340504	2.41103279	4.97380555	-3.1891747	1.51763856	0.77790302	2.07212014
16.2466462	13.4470716	19.3723195	18.1992296	18.1082933	8.29508014	13.9257718	12.3197177	12.8515276	14.3895263
7.52774756	7.1892944	11.8638352	8.07267586	7.95274684	8.57029073	10.9328506	5.32721944	6.58664586	6.12588779
21.870241	23.0760664	22.087068	21.3237499	21.2228553	20.7664993	25.4728987	21.7505873	16.2411165	17.0156962
6.732942	10.1588494	6.41551896	7.59734813	7.28403316	6.49823643	7.49849686	4.39794084	6.76021074	7.01208931
6.38977666	8.14357493	6.65625415	5.39959309	4.78296703	8.41049266	4.635165	5.83979763	3.71474967	8.43088549
8.41199815	5.87917916	4.20514012	5.88894944	6.59126771	40.306979	12.1387736	8.31543861	9.39828605	17.2789013
16.7063092	14.1171	18.5737633	18.0493178	13.2338474	28.2876419	17.4729268	19.7612263	18.808581	23.2064825
-12.306917	-12.645453	-12.996491	-16.839681	-16.380085	-16.50758	-18.26926	-22.248493	-20.174003	-23.905902
7.91284195	6.01394427	5.95591748	5.32365791	3.55793425	7.50838983	6.37435842	5.78608141	4.05167628	6.83878086
14.4997187	16.1132907	12.1265365	11.1843002	14.5837944	9.42230417	11.8193277	11.3073568	8.19862762	9.38303622
17.0448085	20.1051038	14.4760944	15.7159235	17.4455068	14.341617	7.68342186	6.81311439	8.90673495	10.1681686
2.16530156	1.913935	1.89588267	2.66036388	1.44309028	1.04552663	1.13773478	9.36895409	2.66065295	4.67471484
1.39694693	0.96562073	-0.9187668	-2.1598239	-4.309919	-4.5253887	-8.138477	-8.6988213	-7.4829233	-11.78546
7.38992823	5.59836386	8.75618209	4.12726268	5.81665658	62.8288188	2.92434802	2.572389	2.25972749	2.3127648
11.0836586	3.9354714	3.94044659	1.53116496	0.13910531	4.63725773	2.32576385	1.68998471	3.09633785	1.94267969
-2.8874771	-2.989582	-7.7905422	-7.4340218	-8.5665063	0.10525622	-6.9062073	-6.6949145	-6.3368976	-8.6579217
-3.6628211	-5.2521074	-5.3461929	-9.2068963	-9.5336985	-2.7386704	-7.9560011	-6.5884364	-11.154195	-9.4897631

2010-06	2.22158421	-9.8721961	-1.9809965	-1.8159909	-0.8947418	14.133496	-9.3514004	-10.068078	-10.581368	-10.305434
2010-07	1.6771	10.1892843	-0.6352585	-2.9995439	-0.7241254	15.9946207	16.6100312	15.1877355	14.6301056	15.3913034
2010-08	1.57777273	0.96302426	-10.451154	-1.3947338	2.18350757	22.6663355	8.41075539	10.956751	6.92159069	10.0847751
2010-09	1.81278421	-0.7478579	-5.11713	-3.789641	-2.1455613	18.3641256	-3.903016	-0.6013198	-2.9537771	-2.790285
2010-10	1.59955	10.4096651	-0.1505903	-3.0949126	-3.2539829	25.3935908	7.16622569	7.83762084	6.06878183	6.44642923
2010-11	1.71445909	-5.96012	-10.618753	0.21128735	6.8716478	27.9781694	1.58387548	3.37189892	1.80614208	1.9255993
2010-12	2.80312609	-3.6926291	-3.941439	-4.8829279	-2.6998374	17.0069399	-2.9208678	-2.7848117	-2.7245567	-3.2562888
2011-01	3.920145	-5.9530731	1.41398793	-5.6456079	-3.9181039	15.3185478	-7.9207247	-7.766934	-7.7872845	-7.3576516
2011-02	2.51529333	2.79607028	-4.0739371	-1.5883039	-3.6861448	14.5617535	8.88030177	10.6854678	9.34777541	8.77497695
2011-03	1.809	-1.6188296	1.75788671	-2.318536	-0.0720883	16.8870872	-1.1291824	0.88546055	-1.0607691	-1.8589676
2011-04	2.08576316	-3.4499589	-1.6635088	-3.5722634	-0.2015369	14.6886182	-4.3145681	-5.3177691	-5.7614233	-8.2831476
2011-05	2.95300952	-8.9392228	-4.9121285	-2.1078794	-2.0533444	10.7009456	-7.7085264	-8.2540176	-7.4738197	-9.391073
2011-06	4.72191905	-2.318136	-4.7148867	-4.8240409	-3.188135	10.6115622	2.2239343	2.15922541	1.77780219	1.45527354
2011-07	4.35064762	-5.3732355	-7.4544381	-5.3045728	-2.2655443	14.2424544	4.84953022	3.13649112	0.52388506	-1.0510687
2011-08	3.16809565	-7.7586176	-5.8594077	-3.6629022	-0.9016894	13.4848954	0.05293133	-2.3926521	-1.1803352	-2.0249009
2011-09	3.47602381	-12.355079	-2.3502666	-4.0849011	-6.0591757	10.862299	-16.876319	-13.095611	-12.672522	-13.904145
2011-10	3.44755625	0.86611253	-3.7737468	-1.9253534	-3.349166	12.1172983	2.86384321	6.41645349	3.68801727	4.27998663
2011-11	3.32447727	-8.3747193	-5.7603481	-3.3961121	-2.7299758	12.2679028	-2.9672951	-5.167889	-3.657525	-4.9449102
2011-12	2.98919091	-10.251577	0.50958256	-2.9827517	0.39238093	19.3380208	-19.129513	-17.855928	-19.247243	-19.073772
2012-01	4.48974667	-1.2483689	-1.3042991	-5.4622055	-3.5179903	13.8360312	-2.9422071	-4.6925114	-3.5521369	-3.9291159
2012-02	3.43050476	3.678459	-6.5545153	-3.2174768	-4.1979113	13.4872806	13.958004	14.6098484	14.3265311	13.3739401
2012-03	2.47261818	-9.3606945	-4.2908928	-1.728431	-0.5939031	11.9274362	-9.795873	-10.036467	-7.2797908	-9.8511742
2012-04	3.19397059	2.71555069	-3.5214908	-0.4237129	-2.4470077	12.7598358	7.29276625	4.74543575	5.04634823	3.64849254
2012-05	2.15196364	-2.1943076	-3.820438	-1.6181659	-0.7470383	14.0649974	0.54661013	1.55067426	0.80332318	2.9384944
2012-06	2.76981	-7.8485722	-6.4774927	-0.0722923	2.27426719	15.6433526	-5.4316914	-5.3775046	-5.3237783	-5.3312571
2012-07	2.7182	-8.2575319	-3.6783633	-1.5352947	1.68844418	14.9412768	-13.134588	-12.54927	-13.853543	-14.459985
2012-08	2.74311304	-5.1608427	-5.9168667	-3.951262	-4.1507108	17.4424052	5.15631118	4.42639483	4.2908426	4.28072503
2012-09	2.89924	-0.5089416	-1.4568746	-2.2996712	-2.0603362	11.8813261	-0.0596897	0.30869586	-0.6298996	-0.244626
2012-10	2.73762222	-3.6518065	-4.7877604	-5.0154546	-1.9957419	11.8927693	0.78420793	-1.4921753	-1.201549	-1.1494916
2012-11	2.41620455	-8.3762439	0.13191506	-3.2914772	-0.9590221	15.7816744	-11.574202	-13.136173	-13.048771	-12.693483

-10.827979	-8.7952422	-9.3657095	-7.7287739	-9.0468783	-11.627542	-5.1306523	-7.5900159	-10.16077	-10.378008	-11.022776
13.5029401	11.223918	14.3970724	14.4605357	15.3147879	16.5646998	14.8289953	18.1576574	15.0530927	15.9287098	17.3162175
8.26588307	12.2002166	10.2954462	10.0696917	7.41814966	7.81168637	13.8893994	10.3362682	9.27979393	6.81710492	7.44479232
-0.6583258	-2.5927325	-1.3885636	0.37139047	-2.4253542	0.54290039	-1.4326732	0.76276477	0.825818	-2.6250982	0.02392587
6.54529708	9.42730254	6.45818064	7.86435278	8.19695327	8.78193737	10.4794275	9.59428351	10.0818743	7.93727101	8.42987247
-1.9593523	3.32334925	5.30707127	3.47315002	1.45823543	-1.1479546	4.19606542	4.42238437	1.26821208	3.01075005	-2.9862187
-2.6542083	-4.1570917	-1.9258195	-1.6555419	-4.4062666	-3.2565163	-3.9569191	-1.1832135	-1.0587094	-2.9341584	-4.4688106
-6.6801293	-7.0282152	-6.8677912	-8.5540935	-6.3059805	-4.675819	-4.8385182	-6.9187476	-4.1144157	-6.1253365	-4.4362017
9.17466251	7.78386771	9.30925142	9.7596766	9.12873563	9.29140342	11.1746041	10.9227618	10.0115148	10.0247675	8.88496699
-1.4928384	-3.1233687	-1.3603975	-2.4491663	-2.763037	-0.0809263	-3.5610946	-2.5293302	-2.7223327	-2.4798969	-1.6480554
-9.0231808	-4.1259972	-2.8439873	-4.2923153	-5.7827319	-5.8501725	-1.8845953	-6.0385816	-2.7795239	-6.9476793	-3.3781453
-9.1406837	-5.0241923	-6.1062898	-7.1261151	-7.2753061	-8.2074251	-5.3945966	-5.6928991	-6.9875324	-7.8346357	-8.8454232
1.49231697	3.59735646	3.82520532	4.79243712	3.23672957	2.18436706	4.25938619	3.64595329	4.33362162	4.81814708	2.53815165
1.00938613	2.84177506	4.32658515	1.91445213	2.25441418	1.00065732	4.71077964	3.68035209	3.5227894	1.81052013	2.79973087
-0.4741178	-0.5519511	-0.01037	-1.4946527	-2.1573059	-4.2651613	-3.5663503	-1.8542343	-0.8902443	-3.5265347	-5.0063413
-13.027195	-11.520286	-11.342865	-11.529156	-13.00053	-11.848784	-11.776905	-12.796321	-12.100205	-12.971631	-14.291425
3.40284431	4.6190102	5.31952979	4.40799544	3.85238317	2.84791372	5.61148639	6.36316875	5.37042798	3.91846157	4.13802144
-3.2857148	-3.9396841	-4.1474948	-4.2175274	-5.4034143	-4.8419069	-3.6394733	-0.6951722	-3.7591056	-3.6562314	-3.720624
-16.558851	-17.613656	-14.039327	-16.065583	-17.078974	-17.138814	-15.849627	-14.114615	-14.761141	-17.201781	-16.083143
-4.0079477	-1.057532	-2.1144112	-4.5377998	-3.0547653	-1.6880912	0.17514145	-2.0742674	-1.2112136	-1.7889806	-1.5678552
14.8407328	14.917129	16.1470713	15.3500975	14.47525	12.6914953	14.5820687	13.6326866	13.2128882	13.1708008	12.9662978
-9.1791043	-9.1588162	-7.8014657	-6.8392607	-8.6229114	-9.0984292	-6.8104216	-3.6027037	-6.2254706	-7.4292422	-8.9463107
1.14958112	9.95512369	8.65160824	5.4465356	4.75295139	5.88215531	9.75744462	6.27464605	8.35987463	7.89447166	7.06916603
4.02873959	2.94513624	2.57595881	2.1307283	2.14455059	0.83727486	2.07749141	2.00955365	2.43164335	3.1508653	0.40472098
-4.7331611	-5.3313623	-5.4448873	-6.3684828	-3.8196915	-7.1015162	-6.215466	-7.2359209	-4.7523878	-5.7876323	-7.5293292
-16.344779	-10.112316	-8.6597457	-10.631005	-9.2661108	-10.327534	-7.2259945	-6.5002816	-7.9597797	-10.273991	-9.264781
2.90820615	4.23041955	4.21030791	4.17637054	3.31276403	2.40255323	3.88281693	6.34952901	2.59044462	2.36049337	3.56378861
-0.916559	-0.0458432	1.16071156	-0.2629652	-1.2099688	-2.3935862	1.26018443	0.03427189	0.29141434	0.81062909	1.63226763
-1.5187698	-0.6866617	-0.4441435	-0.3405857	0.31240024	-0.2177483	0.46698246	1.88844361	-0.4483399	-0.2540051	-0.0443996
-12.239433	-12.243746	-13.098272	-12.39183	-10.432411	-12.38319	-10.945824	-12.20109	-14.1885	-11.258575	-11.90074

-8.0851784	-9.7746736	-8.689588	-7.4773763	-9.7693942	-7.9263414	-9.537807	-8.5446669	-5.4069484	-8.3012195
13.1003588	15.3757883	14.4820863	16.0794785	14.7559352	11.8651186	15.0255764	10.9727698	13.7703542	14.4168159
17.132884	10.6931772	7.52637843	6.44571281	7.91219708	11.6440268	8.46493279	3.90418293	3.53638225	0.79266813
3.34386755	1.22845385	0.45343838	-0.7747916	-0.0247062	6.93643346	5.78807523	0.73864169	4.24730291	0.56630842
11.1432529	9.51019589	10.2050617	8.20659159	10.876121	15.6371691	19.9282663	14.6085236	19.0199721	17.2257962
5.91739433	4.85644384	0.00747212	1.76205735	0.73794026	6.20250037	-2.5824353	-2.2452356	-5.587355	-7.5384212
-1.0768679	-3.5685787	-1.4945678	-4.801349	-1.947886	2.34481415	0.78114504	1.38350356	-0.6328113	0.52767647
-9.3845891	-8.2109912	-5.3657982	-4.8643733	-4.1466283	-3.9258155	-6.6568698	-0.9520891	1.01877063	1.15095532
11.3256939	10.6483546	10.6661421	11.0578561	9.66410484	8.23081574	5.03039767	9.6676148	6.59340211	4.52873601
-0.1802824	-0.1407756	-3.0172838	-1.4551779	-1.7586824	-4.3240331	0.78006362	-3.5232989	0.47789142	2.58869361
-2.3429535	-2.1439613	-3.5158913	-3.7601004	-4.3154624	-1.2922726	-0.3786485	-2.9041751	-2.1606982	0.45598833
-6.0855013	-7.325466	-7.8891747	-5.0285813	-7.9768448	-4.0410586	-4.255073	-7.6571413	-7.0904229	-5.9771539
4.37267259	2.78829851	3.20737463	3.6049737	3.92539689	3.63932867	3.10101244	3.40706754	3.14476033	3.47547382
3.34762215	4.16498828	1.47286706	1.35101006	3.79296709	6.77055497	0.87627775	2.37744901	-0.584491	-2.4380283
-0.4028291	-0.9957134	-5.0952308	-4.4856388	-3.7506486	-0.8770516	-4.307856	-3.197602	-5.713773	-4.5452351
-12.916791	-12.066363	-12.550223	-12.599806	-11.882846	-9.5698117	-7.981692	-12.138436	-8.3015724	-7.7995713
6.05531229	4.1784781	4.44796825	4.49860725	3.91487459	6.41146132	2.4452218	2.07836107	3.11651116	6.24281342
-2.2936741	-3.7932215	-3.1326913	-3.87377	-5.2208746	-0.8685668	-5.3601588	-6.5802559	-6.6638663	-5.5173421
-14.578804	-12.672983	-13.101298	-14.043094	-12.618747	-5.8769602	-9.7352929	-6.6463173	-3.2374888	-3.7532715
-0.5531627	0.90527849	-0.1824063	-0.2762179	-1.8499194	0.92040406	2.67100712	4.95104917	4.97470737	5.78670283
14.2862337	13.8268294	12.893144	13.1519107	12.0707324	7.61157782	10.8435493	9.05476171	5.24646808	5.4778688
-6.3470049	-5.9779413	-6.5150999	-7.2785777	-7.0071781	-5.6773917	-5.0871622	-8.217213	-7.245057	-6.9629477
8.65786917	5.88878416	9.28930408	8.91245163	6.98114955	5.37674141	8.14366275	6.65955887	6.35329268	6.53597226
4.95403064	3.39682788	3.95111332	2.76511853	3.10037518	3.46236004	4.53109053	1.5358376	0.8639171	-0.1811667
-6.3505875	-5.843582	-6.5910451	-4.5136116	-7.2055133	0.31820761	-4.4069671	-4.2721297	-5.2571257	-5.5215034
-6.4397789	-7.0959966	-8.8272776	-7.746845	-11.485179	-2.4005871	-4.3195274	-4.703216	-5.0104991	-4.0648285
3.1216916	1.51169604	1.55479471	1.34179674	-1.3921255	-4.3367754	-1.7903976	-2.6711234	-2.4292263	-5.177669
2.0878757	0.77017964	2.54630938	-0.5252992	0.69597387	4.13621163	1.3773641	4.65895822	0.86175296	5.34394766
-0.7671496	0.30240301	-1.29083	-0.5833817	0.08157622	4.25445882	1.20175452	-2.4732341	-0.4067745	-0.5476594
-10.961958	-11.800393	-12.136864	-10.989781	-9.8381908	-9.3211906	-3.5755185	-7.2526632	-2.5090079	-4.1815097

2012-12	2.36937143	12.673008	0.25646821	0.2249152	0.49412435	17.97627	16.4445727	15.2974448	17.0084668	13.9173702
2013-01	2.102065	2.98960302	-2.1649026	0.3669048	0.3042863	17.6619566	4.16146393	4.13599714	4.15454047	3.8364913
2013-02	2.83779333	-1.5697531	-4.4462063	-1.4778945	-0.2467042	12.5454089	5.11034502	4.43745123	3.67955335	3.50620953
2013-03	2.32762381	-7.6867789	-4.7219257	-2.45559	1.23310275	17.6639698	-3.0583178	-3.741133	-6.0919125	-6.2066217
2013-04	2.46625556	-4.5552662	-4.0724253	-1.894086	0.70509897	12.8277177	-5.3811981	-3.3787408	-5.3929099	-3.8554924
2013-05	2.92112273	6.87686789	-6.9921583	0.82582446	1.43641102	22.6990721	15.0700663	15.9409808	15.8601354	12.1572188
2013-06	6.46817647	-21.465915	-10.061699	-6.5819631	-2.0097688	13.9961744	-15.583056	-15.650608	-16.409636	-17.976456
2013-07	3.37004348	0.39272845	-8.2638763	-1.2506087	-2.2272325	20.4899172	8.72325084	6.59752429	9.10993886	7.37861809
2013-08	3.21937727	3.38270103	-2.0629917	-4.217348	-6.5258405	20.1652601	8.82820559	6.6927448	7.68783977	6.86974023
2013-09	3.15052632	2.25431059	-4.8057645	-6.3921932	-3.0242053	22.5898991	3.4917505	5.48342169	3.70491015	4.13215378
2013-10	3.61572222	-6.419954	-3.4432269	-4.003091	-2.8545139	16.0791332	-2.4061395	-3.2172939	-0.971182	-2.8571559
2013-11	3.88667143	1.35836197	-6.7859136	-4.0786921	-4.8434893	16.2112906	9.90559001	10.6338091	11.0643854	10.2302394
2013-12	3.68339091	-7.6294029	-4.8908255	-3.0089385	-1.6945565	12.9994373	-5.78317	-2.4244093	-3.4049838	-4.6426245
2014-01	3.3991	-4.5970241	-10.327903	-1.7422651	0.66988934	23.7628451	1.74541756	4.25406287	2.65646357	0.22801971
2014-02	2.76733125	-1.8996111	-4.9540252	0.02354126	-4.5859886	22.3948622	3.97679425	5.30303359	3.47614825	4.04782009
2014-03	2.27602381	-5.0677645	0.65625488	-2.7828976	-2.3750485	16.7739517	-2.6732204	-2.7043067	-1.2957892	-2.9741505
2014-04	2.47767619	-3.2453232	-2.1705158	-2.3001459	-0.4021328	12.6090756	-0.2750799	0.00776473	-1.7527444	-2.2885962
2014-05	2.411755	-0.9190624	-5.1575719	-2.5306181	-3.2063523	12.3185642	2.81642771	2.85162001	3.45354275	3.13216098
2014-06	2.69834	0.38137258	-5.4417863	-1.6986423	-2.2954796	13.5629869	6.85941442	5.08567602	4.94382541	5.68762817
2014-07	3.2097913	5.3617301	-0.8518956	-6.3246606	-5.3608044	16.4957328	4.78159325	6.34418718	6.35263347	5.85128407
2014-08	2.93719048	-0.1068944	-7.3371611	-1.2693923	-4.4384691	16.466973	8.61894381	6.97159325	7.76271059	6.47367856
2014-09	2.7825	6.89596939	-7.6986746	-3.0270584	-5.9602573	22.556313	15.0284124	17.9331465	16.6285557	17.0258102
2014-10	2.49501111	-0.4210209	-1.4145357	-2.0617762	-2.3838539	13.8493731	3.12118216	-0.0510163	-1.0414712	0.76836592
2014-11	2.5376	6.47811282	1.646257	-5.2158429	2.66789804	22.8228155	4.58366433	2.85365605	3.51524521	5.02154728
2014-12	3.0243913	8.21335851	17.2132229	-16.20789	5.0098231	43.4012302	-12.359276	-11.692981	-12.845689	-12.111397
2015-01	2.77865	-0.8124724	-16.349022	-0.5883788	1.99399118	29.6904578	9.58207265	9.31585462	8.60389951	6.99472548
2015-02	3.05396	3.19927483	-9.8809838	-0.0215025	1.95666228	18.0316158	8.27297789	8.09770415	5.18509797	6.6321961
2015-03	3.33651818	15.44497	-12.682041	-1.290787	-1.5447084	31.0480462	21.7711458	26.6319648	17.1735822	18.3645194
2015-04	2.28319048	16.6805565	-9.5434495	2.41089595	-2.1626354	37.9934697	16.3958044	18.4513273	12.3527574	14.7620617
2015-05	1.2084	13.836538	-23.947787	14.1003324	17.8987651	77.1892811	31.0847649	31.2531226	25.1814782	21.4728947

14.1011251	18.9636923	16.7747008	16.2068621	17.0702367	15.9026238	16.1417426	18.5324193	16.6969761	17.5606448	16.9731103
3.19516498	5.44473271	7.11746002	7.44105177	6.95449345	5.40686304	7.57826024	7.96442248	7.983572	6.01615973	6.01032457
4.93599144	4.44907958	4.4912165	3.61564066	4.70319774	1.40167949	3.61382492	3.96822536	4.46106829	3.60102372	3.42721452
-6.6985267	-3.4882882	-2.4565892	-1.4052154	-1.7971349	-4.2655479	-2.2811043	-1.2498052	-2.7452491	-4.6948443	-4.9772561
-5.2065422	-3.698406	-2.6841518	-4.0209849	-5.9759237	-5.1566339	-1.8993679	-1.9456401	-1.987656	-4.3643041	-3.7288822
15.8310514	13.8266981	16.8649594	14.8256305	15.5074298	11.2830749	16.5025774	17.0997962	15.1284817	15.1737809	13.8479945
-18.048599	-16.828247	-14.099011	-17.405202	-16.173903	-16.734494	-12.972186	-15.493676	-15.755107	-16.4793	-17.487647
6.72563722	8.10156453	7.48427974	7.79293992	5.59471193	3.32121559	10.1220991	8.27975349	7.94906663	8.53365579	4.82734968
6.26250611	9.45174358	8.82320948	8.43652806	8.19415348	7.22140665	13.8458484	7.05987421	9.10284758	8.91330891	9.00372529
2.97039831	6.61306668	4.84572083	5.31147972	3.1785707	3.83816987	6.03418219	8.2903183	4.43224496	5.01529566	3.0022495
-3.3591221	-0.2711164	-0.2672304	-4.2904564	-2.1374774	-2.4439763	-4.8214303	-1.0300976	-2.7720252	-2.0798029	-1.1855675
7.52537681	10.2956858	11.0892756	12.0260407	10.4496342	6.98886903	9.2777891	9.20550764	8.08532466	8.0557395	9.19319322
-2.8425639	-3.9602209	-3.062359	-2.799005	-3.3579612	-2.1284707	-1.713555	-2.0630868	-2.3800902	-4.0802198	-4.05316
-3.09385	4.84617908	6.66886659	1.57139943	-0.1315721	-2.4924061	10.8547169	6.49997439	3.2573274	-0.2416183	-3.074743
1.71663026	7.05513151	5.49930912	3.49851081	3.4430144	4.02576602	11.9065841	7.19551464	2.89181249	4.18001285	2.92680928
-2.216192	-3.5998363	-1.7694966	-1.6494179	-0.0128559	-1.3165421	-1.7103763	-5.1148874	-2.8880853	-2.3489937	-1.2340797
-1.4710449	1.16259427	-1.2539525	-1.0350069	-2.258172	-0.995685	-0.8169158	-2.1548232	-2.8886888	-0.5802994	-3.4790546
2.33272145	4.95498644	2.31678026	3.83719669	3.5880447	0.40167169	6.67145007	5.8184198	3.81131689	1.96828434	1.66745526
2.49749903	7.6764465	9.52434937	6.39984707	4.7876567	2.8548404	8.36634998	5.91108375	4.04070297	4.57922856	3.14337206
6.67828837	5.16091748	7.19374323	6.01800076	6.8967886	8.76229399	8.31799078	6.13756339	9.03917902	7.29673378	7.79809958
4.25400196	8.58800474	8.00448702	4.78357834	6.78081182	5.83830132	8.02689527	7.24450072	10.3931899	6.32146522	2.93414581
13.4535212	17.1626344	20.0960659	15.6864447	14.257193	15.6552315	17.2828437	16.3156325	14.3075613	14.152129	14.3288267
0.15601835	4.55436704	4.40757136	1.89125746	5.05411523	1.87764713	1.69204958	3.75045602	3.2624499	0.72954776	1.99011349
3.59321267	7.95903799	3.11911746	5.09034119	3.8025438	2.91251342	8.63254315	6.261574	4.96798199	5.1084928	5.13100946
-11.104498	-5.1698174	-11.152327	-9.734205	-5.0449012	-7.0615713	-7.5376414	-5.8398024	-6.2908098	-4.999802	-4.8874758
4.88874353	10.0185435	8.73622769	8.33904363	5.67222218	5.70494366	11.6291058	11.4828059	7.70982808	7.70003444	5.26585877
3.91154396	7.9978565	8.14704699	6.9620079	5.85125017	6.14092084	12.1838067	9.24274865	7.09109637	7.14060111	5.74262745
19.9744465	26.0174781	22.7771428	20.3377683	24.3979686	19.3571492	28.2239595	23.5858591	21.7360143	22.5782278	18.3367663
12.3212355	20.7872573	17.4960591	17.7122985	15.2129463	14.7888974	20.8568114	24.2726227	18.2159172	15.9255363	13.9088317
17.5162523	46.4150755	32.3453237	29.1398373	22.9963235	17.5174374	67.8629565	39.3949228	31.9344068	25.4797853	19.8719717

17.834603	19.6087176	17.9553337	17.3566613	17.3149364	14.4374316	15.4473948	18.8272955	13.3603456	20.4305725
8.18572084	5.71115637	6.62680129	6.60024587	5.12242311	8.26230929	2.72687331	7.74140962	7.01796391	5.93413741
5.24590468	3.73700294	3.48748381	5.72954253	3.13630245	2.25469307	-1.1496862	1.21534821	0.32563626	-0.6234159
-1.2058337	-1.3836308	-2.4966352	-5.3142437	-5.6401902	1.83409443	-3.4797827	-6.0671277	-3.1443232	-6.2065788
-1.8240992	-1.3653538	-1.2291508	-0.4170266	-3.5687437	0.39483824	-1.8684789	-0.7435708	-2.4141348	-2.1664742
18.5922318	16.8751932	15.3173673	13.3432608	12.3739486	9.36999133	11.1502702	10.5077479	11.7915044	5.34975113
-14.108189	-11.927686	-15.846699	-14.098383	-16.694009	-10.006708	-11.911935	-12.439244	-9.6571442	-15.091783
12.267238	10.0185915	7.99552119	6.36792127	3.40629087	7.55645504	3.22204622	5.16984242	2.31670397	0.72807856
12.019001	7.04579606	8.1021095	9.9210257	9.74037131	4.7676054	4.5180356	6.46181945	12.4141622	6.46641756
9.64957461	6.81469239	7.35924715	6.01905098	2.28294708	9.15621303	10.8623223	3.56234756	11.5834968	4.54980937
-3.3135379	-2.8687259	-4.2367383	-3.6967439	-1.2961285	-1.2593971	-1.9626478	0.25169244	-2.0278898	-0.6651597
7.78296971	10.0577744	9.41448316	9.56774631	5.57971242	7.91761872	4.23757578	5.09864265	3.74862635	3.64017821
-2.2964094	-1.6685003	-2.4681189	-3.6108492	-3.0017047	-1.456882	-4.2477417	-2.4774164	-3.4830467	-4.5849308
10.0998773	6.29839485	2.40991829	2.84291575	-2.4317247	5.61344344	0.72674208	-0.3681405	-0.2793641	-4.5633123
8.33347203	5.63912019	5.46619689	1.44077016	1.30589375	0.02438285	3.21519038	-0.3330521	-0.7156085	3.74348429
-3.8826778	-5.611648	-3.8912674	-0.9796777	-2.2967504	-6.5480237	-4.390601	0.75261354	-2.4245352	-0.6524002
-1.9462143	-0.7549679	-2.3939101	-1.8064719	-2.2085878	1.49585956	0.05173859	0.93568173	-1.0799059	1.38634398
4.37650636	3.88315222	3.39405773	2.042001	0.79672471	2.59839134	2.48136274	0.79535128	1.58863997	0.22048977
6.1914303	6.79239709	3.42444821	3.58374484	3.02378309	4.8523965	1.7776024	4.00850779	1.36689483	2.00648913
9.97335289	6.7605471	8.05491397	9.35136844	9.83339103	6.25072325	7.87504208	7.56051385	8.98801976	9.72797714
7.25552506	7.26628222	6.50220302	5.01772561	4.51332733	6.2819393	2.82789692	2.44177119	1.84423064	0.09875159
15.6120344	11.7173239	13.1436612	11.4856197	10.7299225	12.5213513	6.29077668	8.02597221	8.31280746	4.88102421
1.20693042	1.99153811	2.54679333	1.56660644	3.13034115	2.1248566	1.26858339	2.15666018	3.73759567	3.74546645
4.38429397	4.04924054	8.13787081	5.71044096	7.31545959	8.41200896	10.7334192	18.0396105	13.6988927	13.1406757
-4.0958816	-2.3511007	-2.4137731	-1.8784547	4.784113	11.4434746	25.8231374	21.1070216	22.4802324	31.5546303
10.6120659	11.3706241	9.36760135	6.70091518	3.39393394	19.5953868	3.74638211	9.1582795	3.24655515	-5.4851381
9.12647049	12.0991736	8.33623062	8.34887367	5.88363586	13.1806708	7.49655428	6.05406227	8.83597716	3.94301156
28.843467	27.0784566	23.4929972	24.7685497	20.5925201	24.4841604	22.0728593	15.0958088	16.6708691	14.7413999
29.8745006	18.1191475	18.9453967	19.5222627	17.0612601	31.6226623	20.5403814	21.0333575	22.3396839	19.7002708
63.2507412	38.9056837	31.2595871	22.7549535	22.8341473	33.0160863	16.7147213	13.9300774	19.1323303	2.34423659

2015-06	1.163	-11.683916	-0.3702514	-7.0196444	-10.515118	41.7387744	-6.1025639	-10.707167	-8.658938	-15.324369
2015-07	1.27677391	-17.048639	-5.5112627	2.38967606	5.31629503	29.0055519	-13.293361	-21.057104	-21.045031	-25.136157
2015-08	1.6847619	-17.03121	-3.2933448	-3.3215165	-4.5868629	28.2519313	-14.372045	-16.897486	-15.775436	-15.785489
2015-09	1.909205	-7.8285782	-5.5129334	-0.5085427	3.28845891	20.3853373	-3.660678	-4.3571224	-3.5273414	-8.0013849
2015-10	1.87482353	15.226817	-11.947641	0.15637252	2.71994298	32.3651728	22.3867528	22.3502902	22.2010707	18.429933
2015-11	1.78571429	4.42180789	-11.623969	-0.5998976	4.89574661	35.07314	17.7829086	12.4608243	11.0735746	8.09447983
2015-12	1.84078261	3.49391225	-3.2739444	-2.6790498	-2.8449202	27.3454727	9.75137859	14.0216476	10.41452	9.53806058
2016-01	1.97875	-29.535931	-0.2280477	-2.6121075	-0.4222616	22.1664057	-24.511469	-31.835864	-33.176358	-32.287243
2016-02	1.9756875	-4.6618788	-2.325923	-2.6765531	-0.7741473	16.3611425	-2.4973464	-3.2469453	-2.9772397	-2.0749827
2016-03	1.97373913	14.4564863	-7.0826042	1.24265547	-1.3537869	24.579885	20.3277034	21.2673705	22.0782276	19.9791088
2016-04	2.008915	-3.8690243	-2.2562503	-2.5908048	-0.0657995	16.9805211	2.70138221	4.17351867	2.05708165	2.31457239
2016-05	2.0033	-2.4667161	-3.2985683	-1.0559933	-0.767064	13.9590653	-4.5043789	-1.4403917	-3.1734546	-3.6372509
2016-06	2.0145	1.4554851	-7.6778685	1.56841083	0.71059622	19.9250154	8.63524354	5.9338721	5.71508328	3.93630287
2016-07	2.00947619	-1.4825983	0.04584907	-3.5943321	-0.4715828	18.6767463	-0.8793953	-1.8851431	-2.7555998	1.47173285
2016-08	2.0252	2.70060574	-5.7604152	0.45465522	1.9751661	18.3875288	6.81734263	4.80964833	6.22856874	5.75707114
2016-09	2.134705	-3.8694471	-3.392875	-2.3852764	-1.9679205	13.6313732	8.70007761	1.76332247	2.0483764	-0.0492998
2016-10	2.2003125	0.95411635	-4.5980468	-1.9003475	-2.9540152	12.5238623	8.3996628	6.1684469	3.07528364	3.03460205
2016-11	2.25123636	1.68969515	-1.7548455	-3.2961233	-1.207011	16.5159732	17.7895032	2.57880012	5.08559784	3.08285026
2016-12	2.30028636	-7.6598633	-3.4003383	-2.7255997	-3.1367463	17.042509	3.76505593	0.36609437	-3.9534303	-1.6236834
2017-01	2.20131111	-2.4544093	1.12264385	-3.9364103	-3.03659	14.629154	1.77113803	-0.1299396	-5.1320523	-4.828109
2017-02	2.35459444	0.80334089	-9.5840381	-0.9605159	2.79308841	14.1927419	15.7297941	10.3324483	5.85638483	2.67682549
2017-03	2.48920435	-3.0714367	-5.443571	-2.2827094	3.58701851	15.5367279	5.14425775	-3.4913456	-5.7767373	-2.6596327
2017-04	2.56428333	-5.50116	-2.2525845	-1.6633416	2.33301139	17.4302251	-6.296727	-10.611982	-6.2631038	-9.885513
2017-05	2.74593	-5.5845793	0.30527531	-6.0913021	-1.0420832	18.4012973	-4.6689357	-10.200337	-10.039827	-11.36883
2017-06	2.77741818	2.02510319	-5.0336404	-1.5815984	0.83480165	12.0566754	4.38162384	2.40652297	3.56914001	3.25506512
2017-07	2.67962857	-1.0658062	-0.8040894	-4.0439795	-4.7592894	17.9107345	-5.6876912	-5.7118881	-6.7916044	-6.7240963
2017-08	2.82102609	0.22914695	-7.7401667	0.09740714	-0.0651022	14.7549317	9.25942868	6.75394254	6.25974855	3.5169883
2017-09	2.74206667	-1.4630005	-7.4625592	-0.589378	3.58953265	15.9205761	4.462599	0.73546847	0.31570552	0.92338002
2017-10	2.66019412	-1.5545226	-4.8635438	-2.9875087	3.20075938	15.9076525	-0.5397906	-2.1597381	-1.6225785	-4.5426108
2017-11	2.72791818	-5.6793928	-0.9554311	-4.1762535	-2.4082144	15.6401074	-8.108237	-10.294107	-10.528781	-10.27468

-11.897112	-10.587337	-6.9265483	-11.770119	-9.862032	-11.213053	-7.159666	-4.4042564	-6.6575538	-4.7089052	-9.6166576
-26.209416	-13.974575	-15.642646	-19.960354	-20.628853	-21.727801	-3.9784842	-15.357402	-17.364983	-17.470453	-18.776303
-14.793712	-10.600936	-13.43876	-12.028637	-14.242652	-15.206868	-11.414993	-12.561782	-14.821658	-10.572601	-13.832989
-4.7026568	-1.3813458	-2.3042803	-2.7011258	-2.6991933	-4.7380579	3.82697335	-1.2605513	-4.3399792	-4.1808792	-3.8003882
19.1186909	23.2691556	26.3007718	24.1627708	17.8169431	17.9297432	27.0476969	19.8578863	26.7532758	21.056041	20.1083585
8.25747177	18.5415086	16.2795523	12.1304351	8.63725235	8.50175261	22.2606961	15.3719805	14.4410279	7.83828738	8.27554679
7.15043764	11.001478	10.3612364	10.0127629	7.84084255	7.50395933	10.568674	5.58406164	4.49475606	9.1560919	6.27261525
-31.124067	-22.821539	-26.866698	-26.753029	-29.919822	-28.65481	-23.978322	-30.226419	-26.945471	-29.307175	-29.017975
-0.2429996	-0.2409225	-2.6565436	-2.3712938	-3.6716189	-2.3840486	-1.6498633	-2.5055532	-1.0554943	1.43177221	-3.1342789
16.0343897	19.8939223	20.0021713	18.909679	17.2259502	17.1130249	22.0453265	20.0485528	19.9398909	17.7482658	16.4256817
1.20360231	2.46364412	3.95708093	0.1839956	2.59100536	-1.8140672	-1.1480762	2.49156669	-0.3730371	-0.7575612	-0.1871239
-6.3456975	2.20242511	-0.8430526	-1.1866043	-3.4649167	-2.9863564	1.68015339	2.22330357	1.75777649	-0.5062511	-2.4550212
4.7493255	13.956586	7.45167018	5.80997278	5.4516598	6.96714518	9.26287751	10.0934651	7.52263978	6.84759966	3.3096264
-0.8592774	4.96896226	-2.7478308	-1.3536455	-0.7880699	-1.7483412	9.98162457	1.5273417	1.03574427	-1.4037875	-0.7374022
5.49032124	13.2571537	4.4387453	4.34036322	5.45405346	5.59631117	12.1753067	4.59119181	5.97945969	4.63691513	4.25942656
0.20680677	0.2799326	1.23333801	3.48612786	-0.2024319	0.37132621	-1.3500991	-0.5934639	-0.0710253	-1.7732356	-0.9836258
3.82003203	8.79646246	12.6839219	4.11398197	2.94558404	4.89085916	8.59592102	4.89477787	3.69480028	4.93519135	3.11736802
2.26524426	11.4694003	7.55265691	3.44153166	2.83060567	2.30457259	6.12576021	2.0976626	2.23516841	2.8656694	5.84063647
-3.4036585	-2.6132036	-5.9001853	-7.0257139	-3.6512112	-3.8807724	-1.9306024	-3.6092542	-4.7095772	-3.4335607	-3.7501175
1.12221168	-0.2645575	-1.4777969	-2.6730831	-2.9308132	-3.3775113	-2.1370189	-2.3304631	-3.7379833	-2.1640629	-2.1189516
4.85494589	13.2072206	7.57052761	5.15088311	4.0025088	3.64020406	10.0902551	6.0894632	5.48311138	5.00307292	3.26915652
-6.3660713	3.54001675	-1.6968628	-1.2636837	-2.8469451	-3.1749129	-1.4109359	0.83555678	0.12525656	-0.3229602	-2.2994146
-11.617489	-4.7548789	-7.690004	-7.5590942	-7.5809958	-8.7388739	1.46564872	-3.9041147	-4.1059169	-6.3939591	-7.1183445
-10.747402	-5.5189513	-7.2455895	-8.021789	-8.2061971	-8.7597296	-4.1261574	-6.4812912	-5.6518059	-6.0137011	-8.603495
4.0408911	4.84820131	4.45300019	2.60453727	4.27594418	2.03050137	3.67823627	4.96690424	3.45607192	2.9773852	3.39433981
-2.7643548	-0.9523149	-0.2205215	-1.8403734	-1.2916421	-2.8332944	3.53825687	-1.3679508	0.25098907	1.08838926	-0.7431909
4.2924106	11.6558076	5.79533706	6.27728234	3.89744391	2.30095304	11.9944402	8.92829001	2.42372495	3.11746243	3.36690861
-0.469586	14.6274105	2.52358854	2.13213755	2.40690338	-0.3199182	2.60200995	4.69634429	3.51078001	1.97187784	0.53713666
-4.4980664	0.50465654	-1.193058	-3.2417899	-2.7034212	-1.0933488	4.85383332	-1.1686207	-0.9652053	-2.4834691	-2.0356114
-10.203714	-3.3988707	-4.7013074	-8.7860078	-6.4487756	-9.2944182	-5.656807	-7.045887	-6.7020342	-6.3474219	-7.9493291

-2.9517041	-5.8481117	-8.1263291	-7.0347314	-11.900859	-1.6076575	-3.7213437	-6.7893354	-3.5663672	0.51379398
-11.83705	-10.864813	-12.681664	-14.744702	-13.962968	-9.4026119	-7.7404515	-13.246773	-12.133282	-13.234595
-10.319741	-14.416843	-13.242324	-16.302608	-11.739947	-5.4795109	-9.2008256	-12.754672	-10.633303	-11.108044
0.82490367	-3.4230753	-2.3446201	-5.1933641	-5.8543537	0.89876588	-1.3360586	-4.3507561	-4.5907109	-3.1623219
24.4148995	21.4698841	21.3905288	21.3295629	14.375428	20.7287063	14.5266979	15.2835384	12.6016014	9.48001133
16.3299115	13.7627293	6.61014173	9.97361075	8.76246144	12.5504073	4.81242375	2.22956078	5.55616733	2.10392712
8.66803303	10.2763865	6.63677373	4.43804881	5.60698966	1.98004869	3.40112529	5.19739826	10.6534243	3.7950414
-26.150575	-26.386979	-29.065758	-29.13366	-26.192033	-21.352238	-23.547693	-23.332339	-21.289341	-19.974673
-0.4287529	-2.2749609	-1.2651173	-2.5828663	-2.3525115	1.18852686	-1.50466	-1.3420175	-2.5711019	-0.1498182
20.632477	21.2012062	20.7312694	17.8212695	15.3930531	17.004317	12.9974292	14.5110909	13.4714259	11.8935944
-1.3274104	-2.1196392	5.88990658	-1.1056025	-1.2069499	-1.5367317	-3.2200355	-2.0063128	-3.616838	-0.9914198
-1.1648341	1.7263275	1.19763749	0.7216737	-0.4026003	0.99380809	2.96071762	2.31983694	0.61524902	0.08881351
15.183576	7.68939723	5.92914172	6.38438756	3.61731777	6.98727034	3.90354703	2.15767355	5.4629197	0.4881615
0.13409883	-2.5810713	-0.2592942	1.72634625	0.63936568	0.36240663	2.79354997	3.18293626	4.20147197	3.91369261
15.1942849	4.88460757	2.89515721	6.30441609	4.25257162	11.6723032	3.52925711	3.12546841	9.03212106	4.50506027
-2.9793678	-1.8928513	-2.8637655	-1.4639895	-2.9723451	-1.5324139	-1.4785685	-1.5622262	-1.880242	-2.2397834
4.71792269	4.03470641	5.69086964	3.55327988	2.46524922	4.63749014	2.00625816	4.41582346	3.48767716	3.29611721
2.07570181	1.06555026	2.29824707	2.49952102	3.13172469	8.99010629	1.44341027	5.47465258	7.75120527	7.1481676
-3.6394814	-5.9133126	-2.9721651	-5.0916135	-4.5831754	-4.6174453	-5.1326888	-3.9534018	-5.8534478	-4.990603
-3.9729966	-2.0235712	-1.992492	-1.7669768	-0.4261895	1.71260585	-0.3796888	1.75868669	2.52511291	4.48582749
4.50942946	5.01507792	4.51817276	3.54818113	4.34161797	12.8965778	7.03107109	3.55948987	3.04112018	1.99904664
11.4386996	-1.1441455	-1.8785391	-0.9169529	-1.4726821	2.1297462	4.24649613	2.84552263	0.35166864	0.00832987
-4.1479743	-2.7276425	-3.5338133	-3.1154549	-5.2302517	3.73684915	-0.3978976	-0.515231	1.3897935	-0.2237544
-5.2981156	-6.7134352	-6.0931519	-6.5027966	-6.8150667	-2.7472842	-0.0606168	0.33271365	-1.6088679	1.99558817
4.74574378	5.22281228	4.01501066	4.66245083	4.84392414	7.66178371	8.99877614	4.23727834	3.29887752	3.74881716
1.59200988	3.60034067	1.3692046	1.88847416	3.96050103	3.87036282	3.76704355	4.97937789	5.57566198	4.26761754
9.60638254	4.29169377	4.64334343	4.0204829	1.51361304	7.61628165	6.25662845	3.15789839	5.07044836	2.1821044
5.54561792	3.52779125	2.03659777	1.392179	0.17313689	7.40021451	2.151804	1.92075047	1.8484856	0.07805935
-0.8189639	-0.697383	-1.2919304	-0.7665161	-2.2595929	7.16671715	6.68197916	1.35940499	4.98892035	2.19824945
-5.0948994	-3.4298331	-3.7541223	-2.9229264	-3.4175139	-1.5944631	-1.6233377	-1.321488	-0.4538452	0.31906241

2017-12	2.67252381	-2.9418485	-4.3284772	-3.4919293	-0.5369755	13.1096389	-1.5367944	-3.6637246	-3.794021	-4.1975687

-3.7123751	2.62354356	-2.1902072	-2.1171459	-4.3337602	-3.4736209	0.24447963	-0.9731351	-2.7180419	-2.1754094	-3.3064624

-0.1096195	0.54244766	0.09767522	-1.1271181	-1.0150607	3.03875802	3.22894587	3.45301115	2.52776908	0.68192462